Dpto. de Teoría de la Señal y Comunicaciones
Universidad Carlos iii de Madrid

TESIS DOCTORAL

CONTRIBUTIONS TO THE SOLUTION OF THE RATE-DISTORTION OPTIMIZATION PROBLEM IN VIDEO CODING

Autor: José Luis González de Suso Molinero
Directores: Dr. Eduardo Martínez Enríquez
Dr. Fernando Díaz de María

Leganés, 2016

Tesis doctoral:
Contributions to the Solution of the Rate-Distortion Optimization Problem in Video Coding

Autor:
José Luis González de Suso Molinero

Directores:
Dr. Eduardo Martínez Enríquez
Dr. Fernando Díaz de María

El tribunal nombrado para juzgar la tesis doctoral arriba citada, compuesto por los doctores:

Presidente: d. josé prades nebot

Secretario: dña. carmen peláez moreno

Vocal: d. julián cabrera quesada

acuerda otorgarle la calificación de:

Leganés, a 4 de Julio de 2016

Para Nuria y Sergio...

ABSTRACT

In the last two decades, we have witnessed significant changes concerning the demand of video codecs. The diversity of services has significantly increased, high definition (HD) and beyond-HD resolutions have become a reality, the video traffic coming from mobile devices and tablets is increasing, the video-on-demand services are now playing a prominent role, and so on. All of these advances have converged to demand more powerful standard video codecs, the more recent ones being the H.264/Advanced Video Coding (H.264/AVC) and the latest High Efficiency Video Coding (HEVC), both generated by the Joint Collaborative Team on Video Coding (JCT-VC), a partnership of the ITU-T Video Coding Expert Group (VCEG) and the ISO/IED Moving Picture Expert Group (MEPG).

These two standards (and many others starting with the ITU-T H.261) rely on a hybrid model known as Differential Pulse Code Modulation (DPCM)/Discrete Cosine Transform (DCT) hybrid video coder, which involves a motion estimation and compensation phase followed by a transformation and quantization stages and an entropy coder. Moreover, each of these main subsystems is made of a number of interdependent and parametric modules that can be adapted to the particular video content.

The main problem arising from this approach is how to choose as best as possible the combination of the different parametrizations to achieve the most efficient coding of the current content. To solve this problem, one of the solutions proposed (and the one adopted in both the H.264/AVC and the HEVC reference encoder implementations) is the process referred to as rate-distortion optimization, which chooses a parametrization of the encoder based on the minimization of a cost function that

considers the trade-off between rate and distortion, weighted by a Lagrange multiplier (λ) which has been empirically obtained for both the H.264/AVC and the HEVC reference encoder implementations, aiming to provide a robust solution for a variety of video contents.

In this PhD. thesis, an exhaustive study of the influence of this Lagrangian parameter on different video sequences reveals that there are some common features that appear frequently in video sequences for which the adopted λ model (the reference model) becomes ineffective. Furthermore, we have found a notable margin of improvement in the coding efficiency of both coders when using a more adequate model for the Lagrangian parameter.

Thus, contributions of this thesis are the following: (i) to prove that the reference Lagrangian model becomes ineffective in certain common situations; and (ii), propose generalized solutions to improve the robustness of the reference model, both for the H.264/AVC and the HEVC standards, obtaining important improvements in the coding efficiency. In both proposals, changes in the nature over the video sequence are taken into account, proposing models that adaptively consider the video content and minimize the increment in computational complexity.

RESUMEN

En las últimas dos décadas hemos sido testigos de importantes cambios en la demanda de codificadores de vídeo debido a múltiples factores: la diversidad de servicios se ha visto incrementada significativamente, la resolución *high definition* (HD) (e incluso mayores) se ha hecho realidad, el tráfico de vídeo procedente de dispositivos móviles y tabletas está aumentando y los servicios de vídeo bajo demanda son cada vez más comunes, entre otros muchos ejemplos. Todos estos avances convergen en la demanda de estándares de codificación de vídeo más potentes, siendo los más importantes el H.264/*Advanced Video Coding* (AVC) y el más reciente *High Efficiency Video Coding* (HEVC), ambos definidos por el *Joint Collaborative Team on Video Coding* (JCT-VC), una colaboración entre el *ITU-T Video Coding Expert Group* (VCEG) y el *ISO/IED Moving Picture Expert Group* (MPEG).

Estos dos estándares (y otros muchos, empezando con el ITU-T H.261) se basan en un modelo híbrido de codificador conocido como *Differential Pulse Code Modulation* (DPCM)/*Discrete Cosine Transform* (DCT), que está formado por una estimación y compensación de movimiento seguida de una etapa de transformación y cuantificación y un codificador entrópico. Además, cada uno de estos subsistemas está formado por un cierto número de módulos interdependientes y paramétricos que pueden adaptarse al contenido específico de cada secuencia de vídeo.

El principal problema que surge de esta aproximación es cómo elegir de la forma más adecuada la combinación de las distintas parametrizaciones con el objetivo de alcanzar la codificación más eficiente posible del contenido que se está procesando. Para resolver este problema, una de las soluciones propuestas es el proceso conocido como optimización tasa-distorsión, que se encarga de elegir una parametrización para

el codificador basada en la minimización de una función de coste que considera el compromiso existente entre la tasa y la distorsión, ponderado por un multiplicador de Lagrange (λ) que ha sido obtenido de forma empírica para las implementaciones de referencia del codificador tanto del estándar H.264/AVC como del estándar HEVC, con el objetivo de proponer una solución robusta para distintos tipos de contenidos de vídeo.

En esta tesis doctoral, un estudio exhaustivo de la influencia de este parámetro lagrangiano en distintas secuencias de vídeo revela que existen algunas características comunes que aparecen frecuentemente en secuencias de vídeo para las que el modelo λ adoptado en las implementaciones de referencia resulta poco efectivo. Además, hemos encontrado un notable margen de mejora en la eficiencia de codificación de ambos codificadores usando un modelo más adecuado para este parámetro lagrangiano.

Por consiguiente, las contribuciones de esta tesis son las que siguen: (i) probar que el modelo lagrangiano de referencia resulta inefectivo bajo ciertas situaciones comunes; y (ii), proponer soluciones generalizadas para mejorar la robustez del modelo de referencia, tanto en el caso de H.264/AVC como en el de HEVC, obteniendo mejoras importantes en eficiencia de codificación. En ambas propuestas se tienen en cuenta los cambios en la naturaleza del contenido de una secuencia de vídeo proponiendo modelos que se adaptan dinámicamente a dicho contenido variable y que tienen en cuenta el incremento en la complejidad computacional del codificador.

Agradecimientos

Con la finalización de esta tesis doctoral no sólo finaliza un ciclo formativo, sino que también finaliza para mí un ciclo vital que comenzó allá por 2009, cuando me vine a vivir a Madrid, y que termina ahora, de vuelta en Pamplona y con todas las perspectivas de futuro abiertas.

Es por esto que si me pusiera a agradecer a todas las personas que han aportado algo a estos 6 años de vida, tendría que escribir unos agradecimientos tan largos como la tesis que sigue, pero creo que es mejor tratar de ser conciso, por aquello de centrarse en lo que en este documento procede.

Quiero agradecer primero a Fernando Díaz de María no sólo su trabajo de tutela de esta tesis doctoral, que ha sido impecable tanto desde el punto de vista técnico como desde el punto de vista humano, sino también el haberme dado la oportunidad de vivir este ciclo y haber puesto todo de su parte para que yo pudiera afrontarlo sintiéndome con confianza. También gracias a mi otro tutor, Eduardo Martínez Enríquez, porque todo lo que aquí se detalla se ha cocinado en nuestros míticos descansos para fumar, en los que todas las ideas, por muy expeditivas que fueran, tenían cabida y merecían ser tenidas en cuenta.

No puedo olvidar tampoco a Amaya Jiménez, mi compañera de camino hacia el doctorado, luchadora implacable, pepinera de pro y un gran apoyo para todos esos momentos de debilidad que todo doctorando experimenta a lo largo de este proceso. Gracias también a Luis Azpicueta, por todo lo que me ha ayudado a vivir y por su paciencia y sabiduría en todo lo relativo a la docencia, algo que amo y que gracias a él y sus consejos he tenido la oportunidad de impartir con pasión.

Gracias a mis “maestros” Manuel de Frutos, Iván González, Sergio Sanz y Rubén

Solera, porque desde el primer día en el que llegué aquí han estado para todo lo que me ha hecho falta sin escatimar en tiempo y recursos, laboral y extra-laboralmente hablando, y han sido un ejemplo a seguir para llegar a escribir este texto. No quiero olvidar tampoco al resto de integrantes del GPM con los que más tiempo he compartido: Óscar del Ama, Fernando de la Calle, Fernando Fernández, Miguel Ángel Fernández, Ascensión Gallardo, Alejandro Hernández, Javier López, Tomás Martínez y Carmen Peláez.

Gracias también a Ana Aguilar, Marta Bravo, Lucía Caballero, Guillermo García, Alberto Herranz, Iñigo Mediavilla, Paula Páez, Maria José Vidal y David Zurdo porque han sido mis pilares de vida extra-universitaria, tan necesarios para disfrutar de una ciudad como es Madrid y sus gentes.

Gracias a Paula Unceta, porque ella y sus empujones estuvieron allí cuando me habría quedado a mitad de camino. Gracias a mi familia elegida pamplonesa, Juan Comas, Eduardo Dachary, Yon Luis Gastón, Javier Quel, Víctor Sanz, Sarai Camarzana, Jorge Machín y el resto de ITTSIs y, especialmente, a Elisabeth Esandia, porque sin sus constantes controles de productividad este documento estaría muy lejos de haberse escrito.

Finalmente, gracias a mi madre, Nuria González de Suso, y a mi hermano, Sergio Molinero, porque ellos son mis referentes de vida, el ejemplo de la lucha, de la buena cara ante la adversidad y de saber disfrutar de la vida tal y como viene. A vosotros os dedico esta tesis.

A todos vosotros, y a los que no he mencionado pero que han formado parte durante más o menos tiempo de este proceso: de verdad, gracias.

Table of Contents

List of Figures

List of Tables

List of Algorithms

Acronyms

AVC	Advanced Video Coding
CABAC	Context-Adaptive Binary Arithmetic Coding
CALM	Context Adaptive Lagrange Multiplier
CB	Coding Block
CIF	Common Intermediate Format
CTB	Coding Tree Block
CTU	Coding Tree Unit
CU	Coding Unit (as defined in the HEVC standard)
D	Distortion
DCT	Discrete Cosine Transform
DPCM	Differential Pulse Code Modulation
FDM	Fast Decision for Merge RD-cost
fps	frames per second
GGD	Generalized Gaussian Distribution
GOP	Group of Pictures
HD	High Definition

HEVC	High Efficiency Video Coding
HSV	Human Visual System
JCT-VC	Joint Collaborative Team on Video Coding
IPPP	P-picture prediction structure
IPxB	B-picture prediction structure
ITU	International Telecommunication Unit
MB	Macroblock
MC	Motion Compensation
MD	Mode Decision
MDD	Minimum Distortion Decision
ME	Motion Estimation
MPEG	Moving Picture Expert Group
MRD	Minimum Rate Decision
MSE	Mean Squared Error
MV	Motion Vector
MV_p	Predicted Motion Vector
PB	Prediction Block
pdf	probability density function
PSNR	Peak Signal to Noise Ratio
PU	Prediction Unit
QP	Quantization Parameter

R	Rate
RC	Rate Control
$R-D$	Rate-Distortion
RDO	Rate-Distortion Optimization
RF	Reference Frame
RFD	Reference Decision
SAD	Sum of Absolute Differences
SAO	Sample Adaptive Offset
SATD	Sum of Absolute Transformed Differences
SD	Standard Definition
SSD	Sum of Squared Differences
SSIM	Structural Similarity
TB	Transform Block
TU	Transform Unit
VCEG	Video Coding Expert Group

Chapter 1

Introduction

1.1 Video Coding

In the recent years, video content has experienced important changes related to the quality delivered to the users and also in the way they consume it. HD and beyond-HD resolutions (4k x 2k, 8k x 4k, for example) have become increasingly popular. Moreover, video-on-demand, mobile television services, stereo and multiview capture and display are some examples of how the video content is evolving nowadays. All these services demand efficient solutions to store huge amounts of data and to deliver the same video content at different resolutions.

Although communication networks have also evolved to provide higher capacities, these new requirements concerning video content still pose a major challenge that requires to compress the video signal very efficiently, so it can be stored and streamed reliably according to the highest quality standards.

Since the emergence of the ITU-T H.261 standard [ITU-T, 1990], the video compression problem has been commonly addressed using a block-based hybrid video codec (encoder + decoder), which uses a prediction stage to take advantage of spatial and temporal redundancy in the video signal; a discrete cosine transform to

represent the prediction residual in a more convenient transformed domain; a quantification process that aims to maximize the zero run-lengths; and an entropy coder to efficiently represent these runs.

Beyond the coding techniques, there is need to define video coding standards that allow the encoders and decoders of different manufacturers to properly inter-operate. During the last decades, multiple standards have arisen. Some of the most important examples are the ITU-T H.261 [ITU-T, 1990] and the ITU-T H.263 [ITU-T, 1995], both defined by the ITU-T Video Coding Expert Group (VCEG); the MPEG-1 [ISO/IEC, 1993] and MPEG-4 [ISO/IEC, 1999], defined by the ISO/IED Moving Picture Expert Group (MEPG); and the more recent ones, the H.262/MPEG-2 [ITU-T and ISO/IEC, 1994], the H.264/AVC [JVT, 2003] and the HEVC [JVT, 2013], all of them jointly defined by the ITU-T VCEG and the ISO/IED MPEG, through the Joint Collaborative Team on Video Coding (JCT-VC). Among all of them, the last three standards have been the ones that have reached the largest deployment in the market, being present in a wide variety of devices used in our days.

1.2 Motivation

The block-based hybrid video coding standards rely on a set of flexible coding tools that should be adapted, on a block basis, to the heterogeneous nature of the video content. Thus, the successful selection of the proper parameters and/or coding tools on a block basis becomes one of the key processes of a video coding standard.

Rate-distortion optimization (RDO) is a technique to tackle this parameter selection problem that has been extensively used on the latest video coding standards due to its ability to find near optimal solutions (at the expense of a high computational

cost). In general terms, RDO consists of minimizing a distortion measure subject to a rate constraint. Alternatively, using Lagrangian optimization, this problem can be formulated as that of minimizing an unconstrained cost function which considers two terms, distortion (D) and rate (R), balanced by a Lagrangian parameter λ, whose value needs to be determined.

Thus, modeling the λ parameter of the Lagrangian cost function adds a new selection problem to those that have to be addressed by a video encoder. Several studies have been conducted on this matter, being the model proposed by [Sullivan and Wiegand, 1998] the most successful one because it works properly for a large variety of video contents.

Nevertheless, several research works [Sangi et al., 2004, Zhang et al., 2010, Zhao et al., 2013, Li et al., 2015] have proven this model to fail for certain types of video contents. For example, in H.264/AVC, some inefficiencies of the reference model arise when coding video sequences with high-motion content or, in other words, when the motion estimation (which is the process that deals with the temporal redundancy by looking for the best block-based matches in previously coded frames) is inaccurate. In HEVC, some inefficiencies have been found in video sequences with high percentage of static background.

Thus, the aim of this PhD thesis is to tackle the λ selection problem by designing new models which adaptively modify λ to deal with those situations where the coding efficiency could be improved. Moreover, the design of the new models have considered with special care the associated computational cost, since RDO is at the core of every decision of the video encoder.

1.3 Aims and Contributions

There have been many research works aiming at improving the λ model in H.264/AVC [Chen and Garbacea, 2006, Li et al., 2009, Zhang et al., 2010, Liu et al., 2012, Yeo et al., 2013, Dai et al., 2014] and a few of them (due to its shorter history) in HEVC [Lee and Kim, 2011, Si et al., 2013, Zeng et al., 2013, Zhao et al., 2013, Li et al., 2015]. The most relevant works will be discussed later in the corresponding sections devoted to the related work. Nonetheless, just to put our contributions in context with respect to the previous work, we will briefly describe the objective of our work. Specifically, our work is based on three premises:

- Proposing standard-compliant solutions for both H.264/AVC and HEVC.
- Gaining an in-depth understanding of the type of video contents for which the reference solutions turn out to be inefficient.
- Designing simple solutions that avoid incurring significant complexity increments.
- Providing significant performance improvements with respect to the reference implementations of both standards.

With these objectives in mind, the main contribution of this thesis has been to propose two λ multiplier selection models which, being compliant with either the H.264/AVC or the HEVC video coding standards, are able to adapt to the video content, improving the non-adaptive reference methods and, therefore, improving the coding efficiency of both H.264/AVC and HEVC standards.

First, the main causes of inefficiency of the reference models have been studied in detail for both video coding standards. From the analysis of the reference model of

H.264/AVC, we concluded that inefficiencies are due to inaccurate estimations of the D and R terms in the motion estimation (ME) process, which lead to a poor coding performance. The conclusion of our analysis goes beyond those of previous works on this matter [Sangi et al., 2004, Zhang et al., 2010], where only the inaccuracies of the estimation of R were considered as the source of potential model errors. Furthermore, our analysis also revealed that those estimation errors tended to be more frequent when the block-matching model for ME is not effective. Consequently, we proposed a model in which 3 different λ values are tested for each encoded macroblock (MB) (which is the basic coding unit), making the proposed method MB-wise adaptive over the video sequence.

From the analysis of the reference model in HEVC, we concluded that inefficiencies are found in the mode decision (MD) process (which is the process that decides on the size of the coding unit), which tends to make poor decisions when coding video sequences with static background. According to this observation, we found some features that describe the motion content of the video sequence. Then, we designed a classifier that decides for each frame whether it has a static or a dynamic background. Finally, we designed a regression model that allows us to estimate the λ parameter of the MD process. Therefore, our proposal provides a frame-wise adaptive λ model for the MD process in comparison with non-adaptive previous works [Zhao et al., 2013].

In a few words, the proposed solutions for both video coding standards improve the reference λ models by proposing novel adaptive λ models which account for certain content-related inefficiencies of the reference models.

The results of the proposed methods compared favorably with those of the JM15.1 reference software for H.264/AVC [JVT, 2010] and those of the HM16.0 reference software for HEVC [McCann et al., 2014] and, additionally, with state-of-the-art λ selection methods which use a similar approach, [Zhang et al., 2010] for the H.264/AVC

standard and [Zhao et al., 2013] for the HEVC standard.

To close this section, in the following lines we summarize the contributions of this thesis for each of the standards considered:

- For the H.264/AVC standard:
  - Analysis of the causes of inefficiency of the reference λ model used for ME.
  - Proposal of an adaptive and computationally efficient method to address these inefficiencies at a MB level.
  - Experimental objective and subjective validation of the proposed model.
- For the HEVC standard:
  - Analysis of the causes of inefficiency of the reference λ model used for MD.
  - Search of features that describe the motion content of the video sequence.
  - Design of an effective and computationally efficient static vs. dynamic background classifier at a frame level.
  - Design of an effective and computationally efficient regression model to estimate a proper λ value for the MD process at a frame level.
  - Experimental objective and subjective validation of the proposed model.

1.4 Thesis Outline

This PhD Thesis is organized as follows. In Chapter 2, a brief overview of the video coding problem is given, focusing on the latest H.264/AVC and HEVC standards, followed by a review of the rate-distortion optimization paradigm and the related work. Chapter 3 and 4 describe the contributions of this PhD Thesis to the H.264/AVC and

HEVC standards, respectively. In both cases a comprehensive experimental analysis of the standard RDO process is carried out, revealing the specific inefficiencies in each case. Subsequently, improved RDO methods are proposed and validated. Finally, the conclusions of the thesis are discussed in Chapter 5, which also provides an outline of future research lines.

Chapter 2

Rate-Distortion Optimization in Modern Video Coding Standards

In this chapter, an overview on hybrid video coding [Richardson, 2003] with emphasis on the H.264/AVC and HEVC standards is provided in Section 2.1. The aim is two-fold: i) to briefly describe the different tools available for the video encoder; and ii) to reveal the necessity of the rate-distortion optimization (RDO) process.

Then, a survey of RDO methods is presented in Section 2.2, discussing their motivation and describing the state-of-the-art solutions proposed for both the H.264/AVC and the HEVC standards.

2.1 Hybrid video coding

Major video coding standards since the ITU-T H.261 [ITU-T, 1990] have been based on the Discrete Pulse Code Modulation (DPCM)/Discrete Cosine Transform (DCT) hybrid video codec, which consists of three different stages: a motion estimation (ME) and motion compensation (MC) stage, a transform stage and an entropy encoder

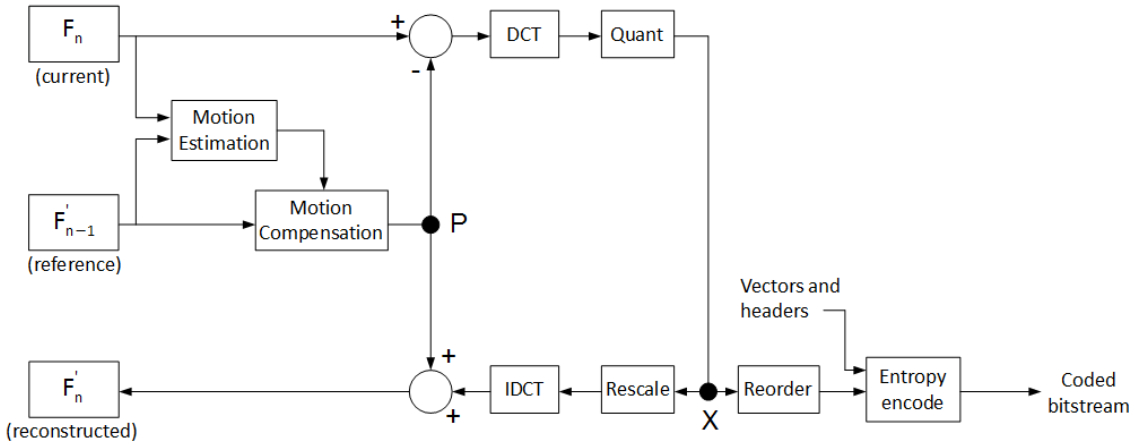


Figure 2.1: Block diagram of a DPCM/DCT hybrid video encoder. Adapted from [Richardson, 2003].

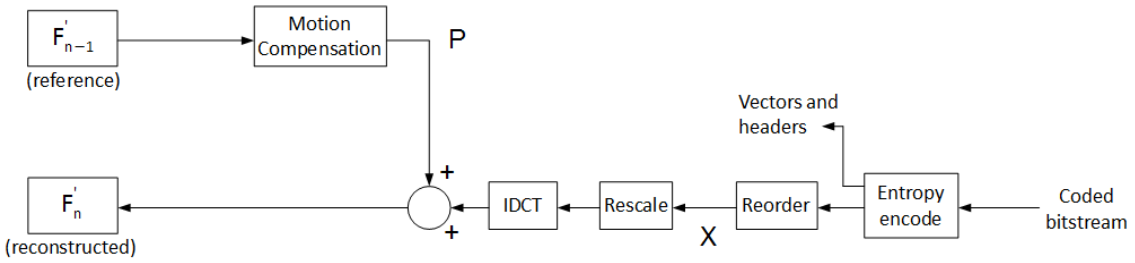


Figure 2.2: Block diagram of a DPCM/DCT hybrid video decoder. Adapted from [Richardson, 2003].

[Richardson, 2003].

This schema was used for both H.264/AVC and HEVC standards, being similar in terms of basic functions, but different in what concerns to details. A block diagram of a hybrid video encoder is shown in Figure 2.1, and that of the decoder is shown in Figure 2.2.

In both video coding standards, the video sequence is processed on a basic coding unit basis. Specifically, the video sequence is divided into frames, the frames into slices and the slices into coding units, which are processed in a so-called *raster* order (starting from the upper-left corner of the slice, moving in the horizontal direction,

and finishing at the bottom-right corner).

From Figure 2.1, it can be seen that a motion estimation is performed for each coding unit using information from a previously encoded reference frame (RF). This reference unit is subtracted from the original coding unit in order to obtain a difference coding unit, with significantly less energy than that of the original, as lower pixel values are present. Then, a transform is performed (in order to gather as much information as possible into a few coefficients) followed by a quantization process. Finally, an entropy coding stage looks for an efficient representation of the data, including motion vectors (MVs), an index referring to the used RF (if necessary) and the headers needed in order for the decoder to understand how the coding unit was encoded. The decoding process, shown in Figure 2.2, performs the same processes (except for the quantization, which is irreversible) in reverse order.

A more detailed explanation of some of these stages will be provided next, making more emphasis on the different solutions proposed in both H.264/AVC [Wiegand et al., 2003b, JVT, 2003] and HEVC [Sullivan et al., 2012, JVT, 2013].

2.1.1 Motion estimation and compensation

The goal of this stage is to take advantage of the temporal redundancy between transmitted frames to compress video data. To this purpose, a predicted frame is built from previously encoded past or future frames and this predicted frame is subtracted from the current one. Thus, the better the prediction the lower the energy of the residual frame.

Taking into account that the video sequence is processed on a coding unit basis, the ME task involves finding a coding unit-sized region in a reference frame that closely matches the current coding unit. Then, the distance in pixels between that

region in the reference frame and the position of the current coding unit is defined as the motion vector (MV). To avoid evaluating all the possible pixels in the RF, a prediction of the MV is made from the neighboring coding units, obtaining a predicted motion vector (MV_p) that points out a reasonable starting position around which a certain area is searched. As a result, the region that minimizes a given matching criterion, known as the *best match*, is determined.

Then, the *best match* region is subtracted from the current coding unit to produce a residue, which is encoded together with an index referring to the used RF and the difference vector between the corresponding MV and the MV_p obtained for the current coding unit.

When considering these processes in the standards H.264/AVC and HEVC, substantial differences can be found.

2.1.1.1 Motion estimation and compensation in H.264/AVC

In H.264/AVC, the basic coding unit is called *macroblock* (MB), which is formed by a 16x16 pixel luma region and two 8x8 pixel chroma regions (when a 4:2:0 video format is used). However, motion estimation (ME) and motion compensation (MC) can be done choosing from a variety of block sizes as illustrated in Figure 2.3.

The ME is performed in a configurable region around the position pointed out by the MV_p for different reference pictures, being the MV_p obtained according to certain criteria from the MVs of already encoded neighboring blocks (one example is illustrated in Figure 2.4). Moreover, depending on whether a P-prediction or a B-prediction is being carried out (which are a prediction based on previous frames or a prediction based on previous and future frames respectively, as it will be explained later), the encoder manages one reference picture list of previous frames, or two reference picture lists of previous and future frames (respectively). This last option

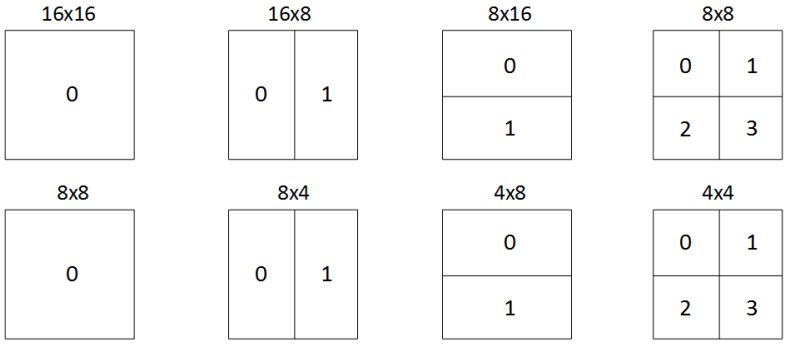


Figure 2.3: Partition modes available in the H.264/AVC standard. Indexes referring to each partition are also shown.

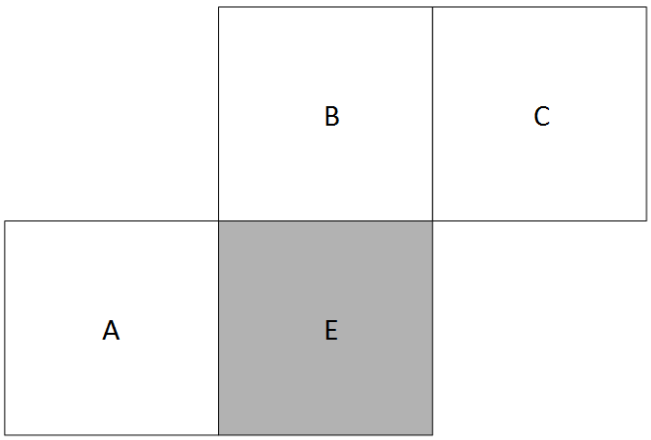


Figure 2.4: Example of motion vector prediction candidates in H.264/AVC when partition sizes are identical to the MB labeled as E. The MV_p is obtained according to certain criteria using the MVs of blocks A, B and C. Adapted from [Wiegand et al., 2003b].

allows the encoder to perform a weighted prediction of the current MB from two different reference frames (RFs). Additionally, ME can be performed with integer-pixel precision, half-pixel precision or quarter-pixel precision, using pixel interpolation in the corresponding region.

2.1.1.2 Motion estimation and compensation in HEVC

In HEVC, the basic coding unit is called *coding tree unit* (CTU). It covers an square region of size LxL (which can be configured to be 64x64, 32x32 or 16x16) and, as can be seen in Figure 2.5, consists of 1 luma coding tree block (CTB) and 2 chroma CTBs. These CTBs form a quad-tree structure of different coding blocks (CB), whose size depend on the depth of the actual quad-tree structure (until a maximum depth, defined in the CTU), being the maximum size the one of the CTB. Then, 1 luma CB and 2 chroma CBs form a coding unit (CU)[1], which is also formed by a prediction unit (PU) and a transform unit (TU). Both the PU and the TU have their root in the CU, and are formed by prediction blocks (PBs) or transform blocks (TBs), respectively, which can be either CB-sized or smaller (by further splitting). A graphical explanation of these definitions can be seen in Figure 2.6.

Thus, each CB can be split according to the quad-tree syntax of the CTB to select an adequate size depending on the current region, generating different CBs of smaller sizes. Then, the PB size is obtained, choosing from 8 possible partition modes shown in Figure 2.7. Comparing with H.264/AVC, HEVC offers new asymmetric prediction modes. For each PB, a motion estimation is performed according to the predicted motion vector (MV_p), which is selected from a set of C potential prediction candidates over a variety of reference frames, with integer-pixel precision, half-pixel precision or quarter-pixel precision. Finally, for each PB, both the difference vector between the actual motion vector and the MV_p and the index for the reference frame are encoded.

[1]In order to distinguish between the general concept of coding unit and the specific one related to the HEVC standard, the latter will be hereafter referred to as CU, while the former will be referred to as *coding unit*.

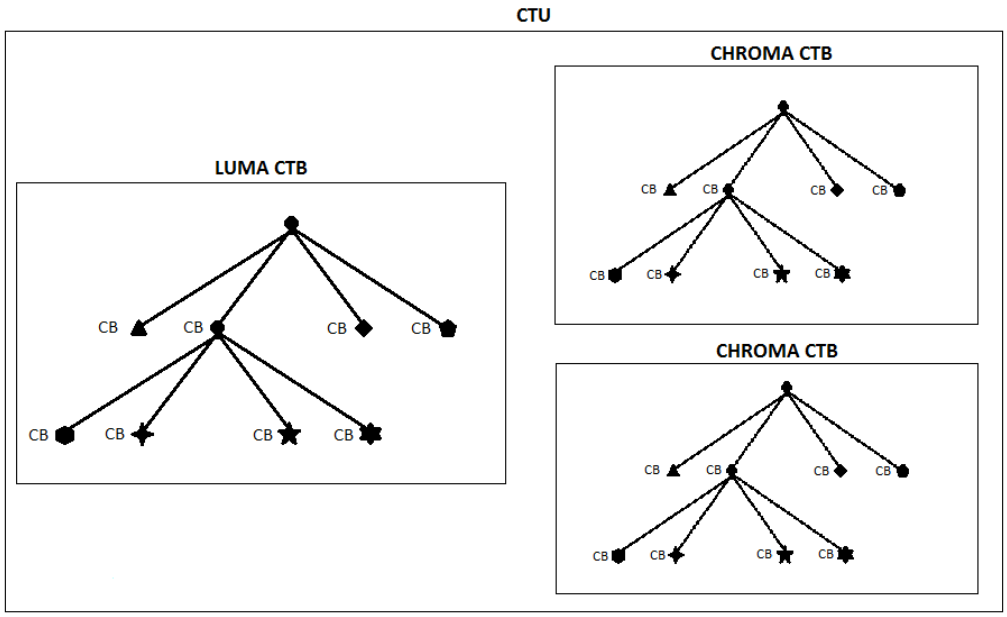


Figure 2.5: Graphical explanation of the CTU data structure. Note that the luma and the two chroma CBs represented with the same symbols form a CU.

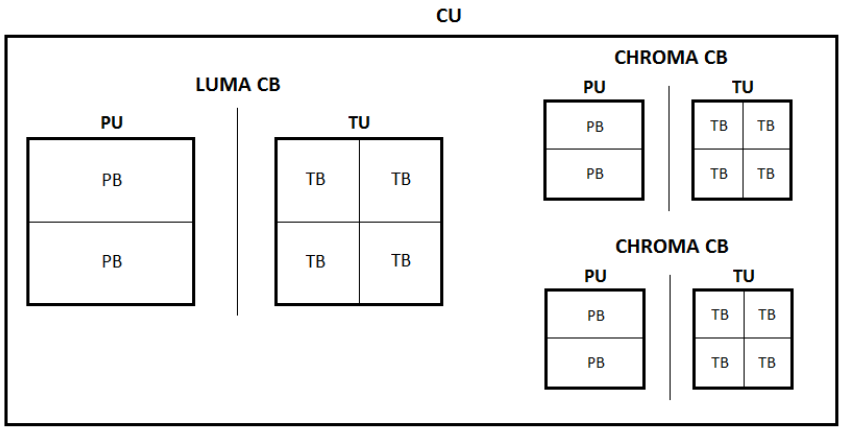


Figure 2.6: Graphical explanation of the CU data structure.

2.1.2 Predictive image coding

The same way the energy of the residual can be reduced by predicting a MB or a CU from a previously encoded frame, a prediction can be done using the previously encoded pixels of the same frame. This prediction takes advantage of the spatial

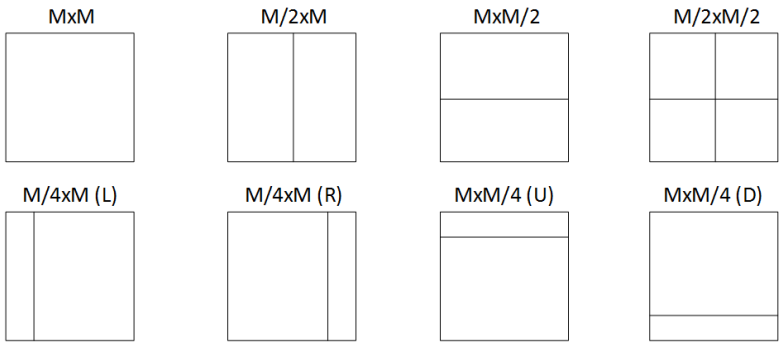


Figure 2.7: PB sizes in HEVC, where M represents the size of the CB. Adapted from [Sullivan et al., 2012].

redundancy within an image to compress data and it is the basis for the so-called Intra modes in both H.264/AVC and HEVC.

2.1.2.1 Predictive image coding in H.264/AVC

Intra prediction uses pixels from surrounding previously coded MBs in order to predict the current MB by interpolation and extrapolation of those. For that purpose, *Intra_4x4*, *Intra_16x16* and *I_PCM* modes are supported by the standard.

Intra_4x4 is used for detailed regions and it allows 9 different prediction modes, the DC mode and 8 directional modes (see Figure 2.8), which will allow the encoder to interpolate (or extrapolate) directional structures as edges within the image. For the *Intra_16x16* mode, 4 prediction modes are supported. Finally, the *I_PCM* mode allows the encoder to send in an efficient way the original MB, just for cases in which prediction is difficult.

2.1.2.2 Predictive image coding in HEVC

Intra prediction in HEVC works according to the transform block (TB) size of the current coding block (CB) and uses the neighboring TB samples to interpolate (or extrapolate). In this case, for every square-sized TB, one mode out of 33 directional

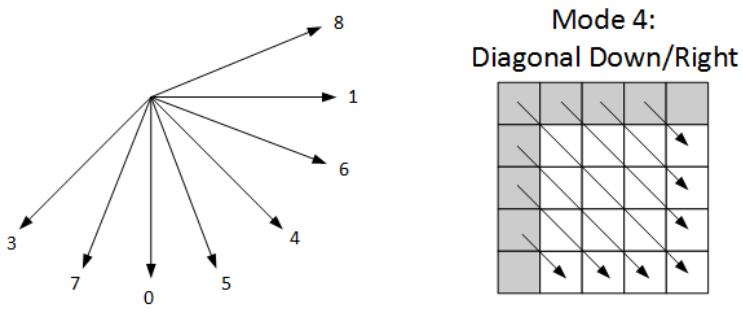


Figure 2.8: (left) Directional prediction modes in H.264/AVC. (right) Example of interpolation using Mode 4. The pixels used for interpolation are represented in gray and the processed coding block is represented in white. Adapted from [Sullivan et al., 2012].

orientations can be chosen, including DC and planar interpolations (which is a surface fitting interpolation).

2.1.3 Transform coding

The main purpose of the transform in a video codec is to convert the residual data into another domain with the goal of obtaining decorrelated and compact data calculated by a reversible transformation in a computationally tractable manner. In the case of both H.264/AVC and HEVC standards, the DCT (Discrete Cosine Transform) is applied to the residual of the motion compensated or spatially predicted coding unit. This transformation tends to compact the energy of the residual around the DC coefficient. Then, a quantization process is performed on the transformed coefficients, according to the quantization parameter (QP), followed by a reordering stage that aims to maximize the length of 0-valued coefficient runs.

In H.264/AVC, the only adjustable parameter is QP, while in HEVC a TU quad-tree needs to be defined out of a variety of possible choices.

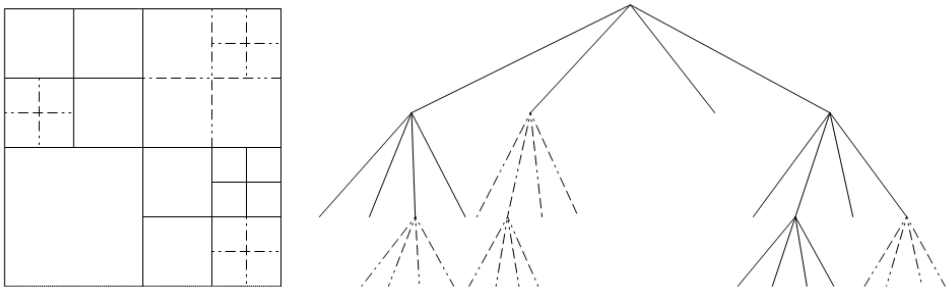


Figure 2.9: Example of the division of a CTB into different CBs and different TBs (left) and their corresponding quad-tree representation (right). Solid lines represents CB partitions and dotted lines represent TB partitions. Adapted from [Sullivan et al., 2012].

2.1.3.1 Transform coding in H.264

Each MB is divided into 4x4 blocks and the transform is applied to each one. After that, a quantization process that depends on QP is performed, followed by a reordering stage using a zigzag scanning over the transformed block.

2.1.3.2 Transform coding in HEVC

The TBs can be recursively partitioned into quadrants in order to reach an adequate TB size using a quad-tree structure similar to that used for ME. An example on how the transform unit can be partitioned is shown in Figure 2.9. After determining the TB sizes, a procedure similar to the one described in H.264/AVC is performed using the QP in order to quantize the transformed coefficients values and a zigzag scanning to maximize the length of the zero runs.

The principal advantage of the transform stage in HEVC when compared to H.264/AVC is that the TU operates independently from the PU, being able to obtain more efficient representations.

2.1.4 Slice/Frame types and temporal prediction structures

In hybrid video coding, there are generally three different types of slices/frames[2] depending on how the redundancy is exploited, namely:

- I-frames: those that are encoded without referring to other previously encoded frames, using the so-called Intra modes only.
- P-frames: those that are encoded referring only to past (already encoded) frames. In this case, the encoder relies on the so-called Inter modes over one list of past reference frames (RFs), in addition to the Intra modes.
- B-frames: those that are coded using two simultaneous lists of RFs, one of them containing past references, and the other containing future RFs. Note that for having access to future encoded frames, the encoding order should be different from the visualization order, as will be exemplified next.

Such types of frames allow the encoder to choose from a high-fidelity high-rate encoding (I-frames) until a lower-fidelity lower-rate encoding (P-frames and B-frames).

Therefore, temporal prediction structures establishing *a priori* how the different picture types will be used, conforming the so-called Groups of Pictures (GOP), are defined in both H.264/AVC and HEVC standards.

2.1.4.1 Temporal prediction structures in H.264/AVC

In H.264/AVC, there is no predefined prediction structures for the standard test conditions. However, we will describe two of the most used ones.

[2]For practical reasons and attending to how the encoding is configured on this PhD. thesis, hereafter it will be considered the slice to be frame-sized, so the term *frame* will be used instead of *slice*.

The P-picture prediction structure (IPPP) is formed by an Intra frame which is sent once in a while (typically once per second), followed by only P-frames which are predicted from the previous frames. This temporal structure allows the decoder to show the decoded frames as they are processed since the decoding order and the visualization order are the same.

The B-frame prediction structure (IPxB, where x is the number of B-frames used) provides a higher compression rate, and is also used in hierarchical structures. But, on the other hand, the decoding and visualization order are not the same in order for the B-frames to have access to previously coded past and future reference frames. In this prediction structure, a prediction over two reference frames is performed either looking for the *best match* in the past and future reference frames through independent ME processes or looking for the *best match* sequentially using the two reference frames jointly, which is more computationally expensive but allows to account for some types of video contents like illumination changes. An example of a IP7B prediction structure is shown in Figure 2.10, indicating the difference between the visualization order (in parenthesis) and the encoding order.

2.1.4.2 Temporal prediction structures in HEVC

In HEVC, some prediction structures are predefined for test conditions [Bossen, 2013], being necessary to chose between either a *low-delay* configuration or a *random-access* configuration. There is an additional *intra-only* configuration, but it is out of the scope of this PhD. thesis.

In the *low-delay* configuration, as in the IPPP configuration of H.264/AVC, the decoding and visualization order are the same, with the ME referring to only past RFs. As a novelty with respect to the H.264/AVC standard, B-frames can also be used in this configuration, being restricted their RFs to past frames. Then, the cases

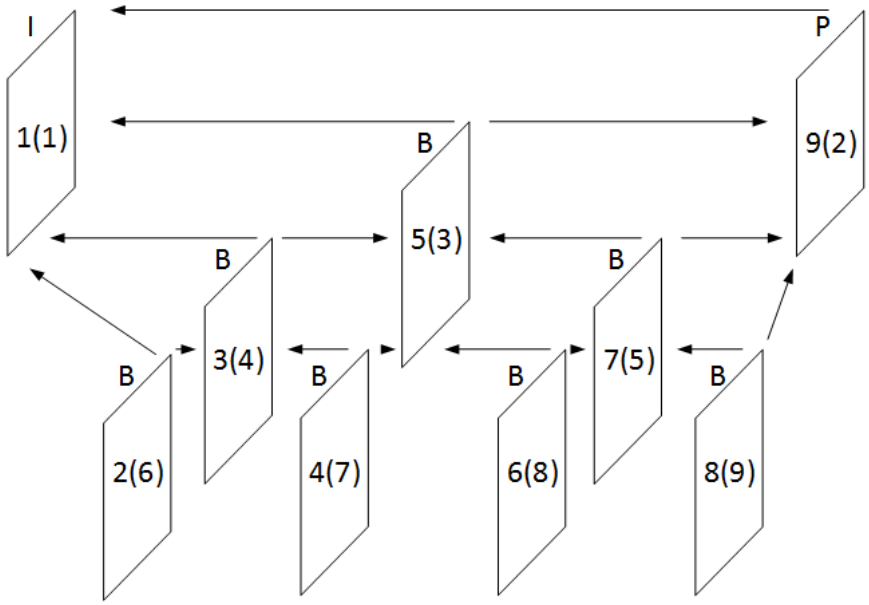


Figure 2.10: Example of a hierarchical IP7B structure. The encoding order (in parenthesis) and the visualization order are shown to make the differences evident.

in which only P-frames are used are denoted as *low-delay-P*, and the others are denoted as *low-delay-B*. For both configurations, a hierarchical QP structure is used to every group of four frames by encoding with a higher QP (which produces less output bits) the frames which are less likely to be referenced. This framework is known as *QP cascading*, and has been proved to not adversely affect the subjective quality [Schwarz et al., 2006]. Specifically, the quantization parameter takes the values $[\mathrm{QP}+3, \mathrm{QP}+2, \mathrm{QP}+3, \mathrm{QP}+1]$ for every consecutive 4 frames with QP being the quantization parameter of the I-frame.

In the *random-access* configuration a hierarchical B-frame structure is used, similar to the hierarchical ones in H.264/AVC but with a predefined number of 7 B-frames and a B-frame with past references only in the lowest temporal layer (substituting the P-frame in Figure 2.10). In this case, the *QP cascading* is performed considering the hierarchical levels. For the lowest temporal layer $\mathrm{QP}+1$ is used, and QP is further

reduced by 1 on each higher temporal layer (reaching to QP + 4 in the non-referenced B-frames level).

2.2 $R-D$ Optimization in Hybrid Video Coding

2.2.1 Motivation

Considering all the coding tools implemented in both the H.264/AVC and HEVC coders, some of them described in Section 2.1, some questions related to the coding process arise:

- How the video sequence should be divided in regions (coding units) for coding purposes?
- How does the video encoder decide between using temporal or spatial prediction for each coding unit?
- In case of temporal prediction:
  - Which reference frame should be used?
  - Which motion vector should be used?
  - Should the encoder just refer to the same region of the previous frame?
  - Which motion vector precision should be used?
- In case of spatial prediction:
  - Which interpolation mode should be used to predict the coding unit?
- Which transform size should be used?
- How fine or coarse should be the quantization?

It should be noted that the more new tools are added to the encoding process, the more new questions arise in order to apply one or the other, or combinations of them. Moreover, it should be taken into account that each answer to these questions has consequences in terms of coding efficiency, as it was explained before. For example, some decisions imply representing the coded video content with more fidelity but, this higher fidelity often comes in exchange for a higher rate. Thus, it becomes clear that coding units that really require higher fidelity should be managed in a different way than those that admit more compression.

These questions and their implications for the coding efficiency make it necessary to design a method which, as optimally as possible, considers the trade-off between distortion (D) and rate (R) with the goal of determining suitable configurations of the encoder to maximize the coding efficiency.

2.2.2 $R - D$ Optimization

The optimization task that the encoder has to face in order to answer the questions posed above consists in determining the most efficient video representation from a rate-distortion ($R - D$) point of view, that is, considering the existing trade-off between both terms.

However, complexity of these tasks becomes even higher due to the fact that the different coding options show different behaviors in terms of R and D depending on the video content (e.g. in a high motion video sequence, using motion estimation seems to be the most efficient option; however, if the block-matching process is not accurate enough, it should be regarded as a better option to perform a spatial prediction through an Intra mode, in order to achieve a better $R - D$ result).

For each of the coding units, each possible combination of the different coding

tools (i.e., MV, RF, block size, QP, etc.), hereafter denoted as $\overline{\theta}$, yields a pair of D and R values. Hence, the goal is to minimize D subject to a rate restriction R_c for the whole video sequence, which is mathematically expressed as:

$$\min_{\overline{\theta}_i} \left\{ \sum_{i=0}^{N} D_i(\overline{\theta}_i) \right\} \text{ subject to } \sum_{i=0}^{N} R_i(\overline{\theta}_i) \leq R_c, \tag{2.1}$$

where N represents the number of coding units in the video sequence, and $(D_i(\overline{\theta}_i), R_i(\overline{\theta}_i))$ is the pair of associated D and R values given a particular choice of parameters $\overline{\theta}$ for the coding unit i.

However, using the Lagrangian formulation proposed by [Everett, 1963], this constrained problem can be posed as an unconstrained one:

$$\min_{\overline{\theta}_i} \left\{ \sum_{i=0}^{N} D_i(\overline{\theta}_i) + \lambda R_i(\overline{\theta}_i) \right\}, \tag{2.2}$$

where λ is the Lagrange multiplier that weights the relative importance of $D_i(\overline{\theta}_i)$ and $R_i(\overline{\theta}_i)$.

Thus, for a given value of λ, equation (2.2) yields an optimal solution $\overline{\theta}_i^*$ for the problem in (2.1) when:

$$R_c = R(\lambda) = \sum_{i=0}^{N} R_i(\overline{\theta}_i^*), \tag{2.3}$$

Moreover, once the constraint has been removed, we have that:

$$\min \left\{ \sum_{i=0}^{N} D_i(\overline{\theta}_i) + \lambda R_i(\overline{\theta}_i) \right\} = \sum_{i=0}^{N} \min \left\{ D_i(\overline{\theta}_i) + \lambda R_i(\overline{\theta}_i) \right\}. \tag{2.4}$$

Therefore, the contribution of [Everett, 1963] to the problem stated in (2.1) is that the global optimization can be solved by finding an optimal solution for each coding unit without considering the global constraint R_c. In other words, (2.4) allows the

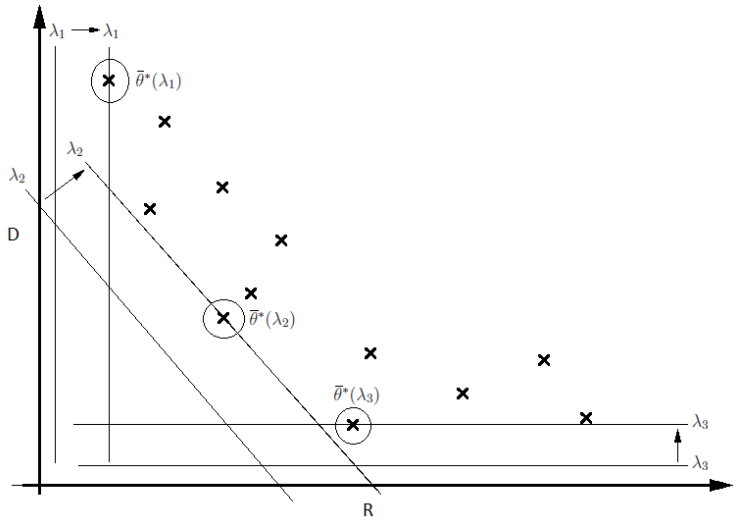


Figure 2.11: Optimal solutions $\overline{\theta}^*$ for three different λ values from a set of discrete operating points $(D(\overline{\theta}), R(\overline{\theta}))$.

encoder to process each coding unit independently, supposedly without considering the solutions obtained for previously coded ones (we will discuss later why we say "supposedly").

Therefore, for each coding unit, a cost function $J(\overline{\theta})$ which should be minimized is defined:

$$\min_{\overline{\theta}} \left\{ J(\overline{\theta}) \right\} \text{ where } J(\overline{\theta}) = D(\overline{\theta}) + \lambda R(\overline{\theta}). \tag{2.5}$$

All this formulation is illustrated in Figure 2.11, where some $(D(\overline{\theta}), R(\overline{\theta}))$ points for an hypothetical coding unit are drawn, forming the so-called $R-D$ characteristic curve [Ortega and Ramchandran, 1998]. Any particular value of λ is represented as a straight line with a given slope of value λ, and the optimal $\overline{\theta}^*$ for each λ will be the one that first hits the corresponding line.

From this example, it can be deduced that the optimal solutions will be found in the convex-hull of the $R-D$ characteristic curve, which is not always reachable. Dynamic programming [Ortega and Ramchandran, 1998] can reach other solutions,

but its complexity is notably higher, specially when there are a high number of coding units, as its complexity grows exponentially with that number.

It also should be noted from Figure 2.11, that the selection of the λ parameter affects the outcome of the optimization task. This means that this parameter should be also determined, along with the optimal $\overline{\theta}$ parameters. This task is quite computationally intensive; consequently, in practical implementations of video coding standards many simplifications are made in order to obtain a more efficient solution.

First, according to (2.4), the independence hypothesis is made so that each coding unit can be optimized independently of the rest. Although this hypothesis is not true because the $\overline{\theta}^*$ chosen for a coding unit actually depends on those $\overline{\theta}^*$ chosen for previously coded coding units, it is a necessary approximation to obtain a practical solution for the optimization process.

Second, in order to avoid testing all the available QP values, some works provided efficient solutions to optimally choose the best QP among an arbitrary subset of candidates, proving that it is not necessary to evaluate all the possible values [Shoham and Gersho, 1988, Wu and Gersho, 1991, Ramchandran and Vetterli, 1993]. Moreover, further studies on this matter [Ding and Liu, 1996, Hang and Chen, 1997, Chiang and Zhang, 1997, Ortega and Ramchandran, 1998, Sullivan and Wiegand, 1998] provided models for R and D as a function of the QP value that allow the encoder to produce estimations without performing the whole encoding process for each possible solution and therefore saving important amounts of encoding complexity. These approximations generated two important improvements in terms of computer savings: (i) Rate control schemes allow the encoder to derive a QP value based on R constraints in an optimal manner and (ii) the λ parameter can be derived from those models and the QP value through the minimization of the cost function in (2.5), assuming that the

$R - D$ curve is differentiable everywhere by computing

$$\frac{\partial J}{\partial R(QP)} = \frac{\partial D(QP)}{\partial R(QP)} + \lambda(QP) = 0, \tag{2.6}$$

which leads to:

$$\lambda(QP) = -\frac{\partial D(QP)}{\partial R(QP)}. \tag{2.7}$$

Other works such as [Le Pennec and Mallat, 2005] proposed specific models for R and D to later derive λ but, among these proposals, the one that has been adopted by the encoder reference models in both H.264/AVC and HEVC standards is the one proposed by [Sullivan and Wiegand, 1998], who empirically derived a relationship that was later theoretically supported based on the *high rate* assumption, which assumes a uniform distribution of D over the quantification intervals when the R term is dominated by the information of the non-zero residual coefficients [Gish and Pierce, 1968]. Specifically, the expression for R as a function of D is:

$$R(D) = a \times \log_2(\frac{b}{D}), \tag{2.8}$$

where a and b are two constants, and

$$D = \frac{(2Q)^2}{12}, \tag{2.9}$$

where Q is half the quantization step. Then, minimizing the cost function (2.5) with respect to the distortion (D) yields:

$$\frac{\partial J}{\partial D} = 1 + \lambda \frac{\partial R(D)}{\partial D} = 0. \tag{2.10}$$

Moreover, by deriving (2.8) and substituting in (2.10):

$$\frac{\partial R(D)}{\partial D} = -\frac{a}{D} = -\frac{1}{\lambda}. \tag{2.11}$$

Thus, by substituting D with its corresponding model in (2.9) and solving for λ, the relationship between Q (or subsequently QP) and the λ parameter is as follows:

$$\lambda = c \times Q^2, \tag{2.12}$$

with $c = 4/12a$.

Third, concerning the motion estimation (ME)-related optimization, which obtains solutions for motion vectors (MVs) and reference frames (RFs), the evaluation of D and R in (2.2) for every potential MV would not be feasible since each evaluation involves DCT-like transform computation, quantization, entropy coding, and inverse processes for reconstruction. The solution to this high-complexity problem consists on simplifying the MV search by using a low-complexity cost function that estimates the selection of the same (MV, RF) pair that would have been selected by evaluating J in (2.5). Thus, the ME process is usually formulated as the minimization of a second Lagrangian cost function denoted as J_{motion}:

$$\begin{gathered} \min_{MV,RF} \{J_{motion}\} \\ \text{with } J_{motion} = D_{motion}(MV, RF) + \lambda_{motion} R_{motion}(MV, RF), \end{gathered} \tag{2.13}$$

where D_{motion} is the sum of absolute differences (SAD) between the original and predicted block for a specific MV and RF; R_{motion} is the number of bits needed to encode the motion-related information; and λ_{motion} is a Lagrange multiplier. Thus,

as the coding unit used for calculating D_{motion} is the predicted one instead of the reconstructed one, large computational savings are achieved.

Once the set of near-optimal MVs and their corresponding RFs have been found by the ME process, they are used to obtain the optimal block size by minimizing J in (2.5), which is referred to as the mode decision (MD) process.

Therefore, in the two considered standards, this MD process refers to the selection of the coding unit size from all the possible choices offered by both intra- and inter- prediction; furthermore, the inter-prediction can use one or two (bi-prediction) reference images.

Finally, considering that the distortion term (D) in J is calculated as a sum of squared differences (SSD) while D_{motion} in J_{motion} is computed as a SAD, an experimental relationship between λ_{motion} and λ was established [Sullivan and Wiegand, 1998]:

$$\lambda_{motion} = \sqrt{\lambda}. \tag{2.14}$$

At this point, given a QP value set by a rate control (RC) algorithm in order to meet a certain target rate (R_c in 2.1) [de Frutos-López et al., 2015], the Lagrange multipliers can be estimated using previous equations (2.12) and (2.14), and the optimal parametrization $\overline{\theta}_i^*$ can be obtained by minimizing J_{motion} (2.13) in the ME stage first and J (2.5) in the MD stage.

These considerations allow the system to obtain a near-optimal set of coding tools $\overline{\theta}^*$ for the coding unit, with a very significant reduction of the computational cost in comparison to the optimal solution.

Once all the basics regarding RDO have been set up, in the next subsections we present some particularizations of the model for both H.264/AVC and HEVC standards. Moreover, we also present a bibliographic review concerning the methods

proposed to improve the RDO process in each one of the considered standards.

2.2.2.1 $R-D$ Optimization in H.264/AVC

Regarding the H.264/AVC standard, the relationship between the Lagrange multiplier λ and the QP that is implemented in the JM15.1 reference software [JVT, 2010] was established using an empirical method proposed in [Wiegand and Girod, 2001], which led to the following relationship [Lim et al., 2005, Wiegand et al., 2003a]:

$$\lambda = 0.85 \times 2^{(QP-12)/3}. \tag{2.15}$$

Nonetheless, other related works have attempted to establish alternative relationships between the Lagrangian λ and the QP. Some of them have attempted to improve the model of λ by making it dependent of the actual video content. One of the most implemented strategies is to make the R and D models dependent of the non-zero quantized coefficients of the residue, which are usually modeled using parametric distributions such as: the Laplace distribution [Lam and Goodman, 2000], the Generalized Gaussian Distribution (GGDs) [Yovanof and Liu, 1996] or the Cauchy Distribution [Altunbasak and Kamaci, 2004, Kamaci and Altunbasak, 2005]. That is the case of [Li et al., 2009], where an algorithm was proposed to accurately select the value of λ by considering a Laplace distribution of the quantized residual and adapting the λ value to the actual video sequence, so that the overall coding efficiency is improved. They model R and D as function of QP, some features of the input sequence, and the frame type. Then, λ is obtained by following the corresponding analytical model. Although the model is elegant, they fail to fully describe the encoder, and some of the assumptions they make in order to simplify calculations lead to specific solutions for specific types of content.

A method described in [Chen and Garbacea, 2006] uses a similar approach, but they proposed a $R - D$ model in the so-called ρ domain, where ρ is a parameter derived from the number of zero-quantized transformed residual coefficients. This proposal has the advantage of leading to linear models of D and R, which notably simplify calculations. Then, λ is dynamically derived from ρ and the estimations of D and R. However, this model decouples λ and QP, making the rate control process more difficult.

Another approach proposed to improve the performance of the reference implementation is to consider perceptual distortion metrics in the RDO model. Specifically, some works proposed using SSIM[3]-derived metrics [Channappayya et al., 2008]. [Wang et al., 2011] proposed a model using the distortion metric $1 - SSIM$[4] to derive a λ multiplier. The same did [Yeo et al., 2013] and [Dai et al., 2014] using a $SSIM$-based distortion measure called $dSSIM$[5] for establishing a relationship between the usual distortion term based on the Mean Squared Error (MSE) and their proposed perceptual-aware distortion, further designing a new perceptual feature-dependent λ model.

Concerning the cost function related to the ME process (J_{motion}), the lack of accuracy of the *high rate* model on low-rate situations, where the number of bits needed for sending the side information (MV, indexes, headers, etc.) is comparable to the rate needed for sending the non-zero transformed coefficients of the residual, motivated some works concerning the λ_{motion} parameter. In [Sangi et al., 2004] a linear model was established for both R_{motion} and D_{motion} to obtain analytically the

[3]The structural similarity (SSIM) index is a perceptual measure used to evaluate the similarity between two signal vectors based on the luminance, contrast and structural correlation [Wang et al., 2004].

[4]SSIM is defined to be $SSIM \leq 1$ and the measure $1 - SSIM$ is defined to be used as a proper distortion measure in the cost function.

[5]This $dSSIM$ measure is another distortion measure which comes from the relationship $dSSIM = 1/SSIM$. Note that, in this case, it can take values higher than 1.

optimal λ_{motion} value, but the method does not provide a significant improvement in performance with respect to the reference model. The Context Adaptive Lagrange Multiplier (CALM) method presented in [Zhang et al., 2010] adjusted λ_{motion} for each block based on its context, that is, based on the Lagrangian cost of its neighboring blocks. This approach has been implemented in the JM reference software [JVT, 2010] since the 10.2 version.

2.2.2.2 $R-D$ Optimization in HEVC

In HEVC, the relationship between the Lagrange multiplier λ and the QP that is implemented in the HM16.0 reference software [McCann et al., 2014] was established using the same empirical method that was used in H.264/AVC [Wiegand and Girod, 2001], leading to the following relationship [Kim et al., 2012]:

$$\lambda = \alpha \times W_k \times 2^{(QP-12)/3}, \tag{2.16}$$

where α depends on the frame coding type and the reference level and W_k depends on the encoding configuration (*random-access* or *low-delay* conditions) and the hierarchy level of the frame within a group of pictures (GOP).

As an alternative to this $\lambda(QP)$ model, other $R-D$ models have been proposed in the literature which yielded different relationships between λ and the QP. Some proposals are based on extending H.264/AVC approaches to the HEVC standard, e.g. [Ma et al., 2012], where the quadratic model for the R and D in (2.8) and (2.9) was adapted to account for the Sample Adaptive Offset (SAO) filter which is a non-linear amplitude mapping to better reconstruct the original signal amplitudes that was introduced in the new standard. Others adapt the model to different distortion measures, as the Sum of Absolute Transformed Differences (SATD) [Deng et al.,

2013].

Another approach to improve the performance of the reference adopted model is to introduce video content adaptation by using parametric distributions (mainly the Laplacian distribution), as in H.264/AVC, but with a main difference. Considering the quad-tree model for the coding unit (CU) and the transform unit (TU), they model the transformed coefficients as a mixture of Laplacian distributions, being independently modeled for each *depth* of the quad-tree structure. This is the case of [Lee and Kim, 2011], where although such models were proposed, they did not propose a model for the λ parameter. This work was expanded by [Si et al., 2013], who used the same approach for modeling λ as a function of the Laplacian parameter and the QP.

Perceptual-oriented RDO has also been proposed for HEVC. [Rehman and Wang, 2012] proposed a model in which the cost function is evaluated with a perceptual-oriented distortion term by using a modification on the SSIM measure to account for the different TU sizes proposed by the standard, but they used the λ parameter in (2.16) to evaluate the cost function. On the other hand, [Zeng et al., 2013] proposed a multiplying factor for the reference λ that depends on the perceptual sensitivity of a coding tree unit (CTU), based on spatial and temporal features.

Other approaches attempted to consider dependencies between CTUs in an efficient way in order to improve coding performance. For example, [Liu et al., 2012] used correlation between CTU residues to model both R and D, and later derive the λ parameter. However, although they claimed that their method is applicable to HEVC, it was not tested. [Li et al., 2015] eliminated the CTU independence hypothesis in order to account for the impact of coding one CTU on the coding of subsequent CTUs, using an approach similar to dynamic programming. To this purpose, they performed a forward motion estimation and evaluated the influence of a certain CTU

in the following ones.

Approaches based on the ρ-domain described in [He and Mitra, 2002] were also proposed since the resulting models are simple due to the linear relationship between R and ρ. For instance, [Biatek et al., 2014] modeled ρ as a mixture of Laplacian distributions and derived a model for R and D, but they did not include the λ modeling. Also in this direction [Wang et al., 2013] proposed a model operating on the ρ-domain in which ρ is modeled as a mixture of Laplacian distributions and which is related with R and, ultimately, the quantization parameter (QP).

A R-λ model was also proposed for HEVC [Li et al., 2014], which was in fact included in the HM16.0 reference software as a part of the rate control subsystem. In this work, they claimed that the R-λ relationship is more robust in the HEVC framework than the typically used R-Q relationship. Thus, they proposed a model in which the rate control acts directly on the cost function through the λ parameter. However, this approach did not take into consideration the *QP cascading* applied to the hierarchical structure of the GOP.

Finally, other proposals designed *ad-hoc* solutions to particular weaknesses of the reference model. Specifically, it has been observed that the reference model tends to be less effective in video sequences that show a static background. Hence, for surveillance video coding, some proposals adapted the HEVC encoder to account for these potential weaknesses. That is the case of [Zhao et al., 2013], who proposed a λ modification based on the percentage of static background in the image for surveillance video sequences. Specifically, classify each CTU into static background percentage bins and then, they find a relationship between the percentage of static background and the optimal λ parameter, which is parametrized specifically for each video sequence in a training stage carried out at the beginning of the encoding process. This proposal yielded interesting results; however, although this performed well for static

and continuous video contents as those coming from video surveillance sequences, it did not work well for general varying-content video sequences, as the parameter training stage for the λ model is performed only once at the beginning of the encoding process. Additionally, [Zhang et al., 2014] proposed a different approach to improve the coding performance of video sequences with static background. They proposed the use of a so-called G-reference frame that intends to model the background and that is used as a long-term reference. However, again, this method is specifically designed for video-conference and surveillance videos.

Chapter 3

Lagrange Multiplier Selection for Motion Estimation in H.264/AVC

In this chapter, we describe the contributions of this thesis to the rate-distortion optimization (RDO) process in H.264/AVC.

First, we analyzed the performance of both $\lambda(QP)$ and $\lambda_{motion}(\lambda)$ relationships, pointing at the latter as the one to have a greater impact in terms of average coding performance. Therefore, our research work focused on improving the λ_{motion} model. Specifically, our study proved this $\lambda_{motion}(\lambda)$ relationship to be ineffective for those video contents that compromise the block-matching based motion estimation (ME) process. Typically, these types of contents include fast and random movements and video transitions such as *fades*, *zooms*, etc. According to our research, in those cases the (motion vector (MV), reference frame (RF)) pair selected by the ME minimizing J_{motion} by applying (2.13) was found to be different from the one that would have been chosen by an exhaustive evaluation using J in (2.5).

Thus, contributions of this Chapter are: (i) an exhaustive study of the cases for which the $\lambda_{motion}(\lambda)$ reference model is not accurate enough; (ii) a new $\lambda_{motion}(\lambda)$

relationship proposal for these cases; and (iii) an adaptive implementation that allows us to apply the new model only when necessary, leaving the reference model unaltered for the rest of the cases. All this work has been described in [Molinero et al., 2011] and [González-de Suso et al., 2014].

The remainder of this chapter is organized as follows: In Section 3.1, the motivation for this chapter is described. Section 3.2 describes the proposed method and all the aspects considered for its design. In Section 3.3, we explain the experiments carried out and the results achieved, which prove the efficacy of our proposed method. Finally, Section 3.4 summarizes our conclusions.

3.1 Motivation

3.1.1 Evaluation of the Lagrangian parameter model for H.264/AVC

As a first step, the Lagrangian model adopted in H.264/AVC [Lim et al., 2005] has been tested to assess its robustness. Both $\lambda(QP)$ and $\lambda_{motion}(\lambda)$ relationships have been parametrized by means of a control parameter which is set for the whole video sequence encoding, choosing from a large range of values and leading to different encoding processes for 6 Common Intermediate Format (CIF) video sequences (352 wide x 288 height size). The goal of this procedure is to find improved versions of these relationships, which produce a better performance in terms of coding efficiency. Furthermore, those video sequences exhibiting significant performance improvements will be studied to look for any common visual feature (motion type, texture, background, etc.) that can account for these improvements.

The coding conditions for these tests are summarized on Table 3.1, where fps is

Table 3.1: Summary of the main coding parameters.

Parameter	**Value**
GOP	IPPP
fps	30
QP values	[20, 24, 28, 32]
RDO	ON
# RFs	3
# Frames	100
c	[0.5 : 0.4 : 2.1]
F	[0.5 : 0.4 : 2.1]

the frames per second and RDO (on/off) is an encoding mode in the JM15.1 reference software which, when activated, performs coding decisions evaluating only the J_{motion} cost function, substantially reducing the encoder complexity at the expense of coding efficiency[1].

Regarding the $\lambda(QP)$ relationship, recalling from Section 2.2.2.1, λ is obtained from the QP using the following expression:

$$\lambda = c \times 2^{(QP-12)/3}, \tag{3.1}$$

where c is a multiplying factor whose value in the reference JM15.1 software is 0.85.

The value of this multiplier has been varied within the range 0.5 to 2.1, in steps of 0.4, comparing the achieved results with those of the reference procedure ($c = 0.85$). To assess the encoding performance, we have computed $\Delta R(\%)$ to measure the increment in output bit-rate for a given output quality regarding the reference procedure and $\Delta Y(dB)$ to measure the increment in objective visual quality considering the luma component Peak Signal to Noise Ratio (PSNR) for a given output rate, using the procedure of curve interpolation based on 4 rate-distortion ($R - D$) points described in [Bjontegaard, 2001]. The obtained results are shown in Table 3.2.

[1]Further parametrization choices involving entropy coding, search range, etc. are later provided in Table 3.8.

Table 3.2: Coding performance results for a wide range of c.

Video Sequence	$c = 0.5$		$c = 0.9$		$c = 1.3$		$c = 1.7$		$c = 2.1$	
	$\Delta R(\%)$	$\Delta Y(dB)$	$\Delta R(\%)$	$\Delta Y(dB)$	$\Delta R(\%)$	$\Delta Y(dB)$	$\Delta R(\%)$	$\Delta Y(dB)$	$\Delta R(\%)$	$\Delta Y(dB)$
Akiyo	1.73	-0.07	0.05	0.00	0.45	-0.02	1.82	-0.08	2.95	-0.13
Coastguard	-0.07	0.01	-0.08	0.01	4.02	-0.24	10.52	-0.60	18.77	-1.02
Foreman	0.89	-0.04	0.23	-0.01	3.74	-0.16	8.92	-0.37	15.30	-0.61
Highway	-2.81	0.05	0.32	-0.01	6.08	-0.12	14.55	-0.27	25.83	-0.46
Ice Age	3.54	-0.22	-0.41	0.02	-2.30	0.14	-4.33	0.27	-5.93	0.36
Nature	6.85	-0.34	-1.02	0.05	-2.55	0.12	-3.27	0.15	-1.38	0.05
News	0.51	-0.03	0.02	0.00	1.58	-0.09	3.35	-0.18	5.41	-0.29
Average	**1.52**	**-0.09**	**-0.13**	**0.01**	**1.57**	**-0.05**	**4.51**	**-0.15**	**8.71**	**-0.30**

It can be seen there that the reference c value is in average robust enough considering different video sequences. The coding performance improvement ($\Delta R < 0\%$ or $\Delta Y > 0dBs$) achieved by $c = 0.9$ is not significant enough.

Although *Ice Age*, *Nature* and *Highway* video sequences show notable improvements over the reference c value, reaching -5.93% bit-rate savings (0.36 dB of quality gain) for *Ice Age* when $c = 2.1$, -3.27% bit-rate savings (0.15 dB in quality gain) for *Nature* when $c = 1.7$ and -2.81% bit-rate savings (0.05 dB in quality gain) for *Highway* when $c = 0.5$, in the rest of the video sequences, no improvement was achieved by evaluating λ different from the reference one. Moreover, in terms of average coding performance, results show that the best option is to apply the reference model. This behavior is to be expected, as the $\lambda(QP)$ model was designed to perform robustly in average over all kinds of video sequences. Therefore, we did not expect to achieve better performance by acting upon the reference $\lambda(QP)$ model.

The same procedure was applied to the $\lambda_{motion}(\lambda)$ relationship in order to assess its robustness. In this case, the relationship is altered with respect to the reference (2.14) by means of a multiplying factor F:

$$\lambda_{motion} = F \cdot \sqrt{\lambda}, \tag{3.2}$$

Table 3.3: Coding performance results for several values of F, relative to that of $F = 1$.

Video Sequence	$F = 0.5$		$F = 0.9$		$F = 1.3$		$F = 1.7$		$F = 2.1$	
	$\Delta R(\%)$	$\Delta Y(dB)$	$\Delta R(\%)$	$\Delta Y(dB)$	$\Delta R(\%)$	$\Delta Y(dB)$	$\Delta R(\%)$	$\Delta Y(dB)$	$\Delta R(\%)$	$\Delta Y(dB)$
Akiyo	0.27	-0.01	0.12	-0.01	-0.02	0.00	-0.01	0.01	-0.17	-0.01
Coastguard	0.26	-0.01	-0.48	0.03	-0.10	0.01	-0.29	0.02	-0.29	0.02
Foreman	0.67	-0.03	0.02	0.00	-0.17	0.01	-0.12	0.00	-0.16	0.01
Highway	0.43	-0.01	-0.16	0.00	0.39	-0.01	1.44	-0.03	2.40	-0.05
Ice Age	1.06	-0.07	0.28	-0.02	-0.65	0.04	-1.74	0.11	-2.61	0.17
Nature	1.18	-0.06	0.06	0.00	-0.29	0.02	-0.32	0.02	-0.80	0.04
News	0.15	-0.01	0.07	0.00	0.07	0.00	0.06	0.00	0.07	0.00
Average	**0.57**	**-0.03**	**-0.01**	**0.00**	**-0.11**	**0.01**	**-0.14**	**0.02**	**-0.22**	**0.03**

which allows to parametrize changes in the relationship. Note that when $F = 1$, the reference relationship is used.

This F factor was varied following a similar procedure as the c value before (see Table 3.1). Table 3.3 shows the results of using different values of F with respect to that of $F = 1$.

In this case, the average coding performance tends to improve with F, reaching a maximum at $F = 2.1$, where -0.22% of bit-rate savings is achieved (or, alternatively, a 0.03 dBs increment in terms of luma PSNR). This behavior is different from the one found by acting upon the $\lambda(QP)$ relationship, which proved to be robust in average.

Thus, although the increment in coding performance is low, this preliminary result points out that the average performance could be improved by acting upon the $\lambda_{motion}(\lambda)$ relationship. Therefore, a further analysis was done on the $\lambda_{motion}(\lambda)$ model in order to find the reasons why it could be not accurate enough and improve it accordingly.

3.1.2 Accuracy of λ_{motion} estimation

Our analysis started by investigating the cases in which the estimation of λ_{motion} given in (2.14) could be improved. To that end, instead of modifying the $\lambda_{motion}(\lambda)$

relationship with an specific value for encoding the whole video sequence as before, the encoder was modified to test several values of λ_{motion} for a given value of λ on a macroblock (MB) basis. Specifically, for each MB, each value of λ_{motion} produces a candidate pair (MV, RF) resulting from the optimization of J_{motion} and each candidate pair (MV, RF) is tested on J. In this manner, the decision on the best pair (MV, RF) is made using the actual R and D values, instead of estimates. As a result, an optimal pair (MV, RF) and, consequently, an optimal value of λ_{motion} are selected. Thus, in those cases in which $\lambda_{motion} = \sqrt{\lambda}$ is the best solution, this approximation is proven to be accurate, and vice versa.

Specifically, 21 different values of the previously defined F factor in (3.2) were tested:

$$F_i = i \times \Delta F, \text{ with } i \in [0, 1, ..., 20], \Delta F = 0.2. \tag{3.3}$$

Hereafter, λ_{motion} will be referred as the value obtained by applying (2.14), λ_i as the product $F_i \times \lambda_{motion}$, and λ_i^* as the optimal λ_i value (the one associated with the optimal (MV, RF) pair selected in J). Note that as the F factor is altered, this ultimately leads to change the balance between R_{motion} and D_{motion} in the J_{motion} cost evaluation:

$$\widehat{J}_{motion} = D_{motion} + (F_i \times \lambda_{motion}) \times R_{motion}. \tag{3.4}$$

On the one hand, $F_i = 0$ produces a MV that minimizes D_{motion} without any rate considerations. On the other, the higher F_i, the more the decision depends on R_{motion}, in detriment of distortion considerations.

The algorithm that selects the optimal value of λ_i^* in a MB basis as described before will be used as an ideal reference (benchmark algorithm), as it explores a wide range of values for F_i and selects the best. It is summarized in Algorithm 1. It should be noted that F_i is evaluated only in the 16x16 mode. This strategy is

Algorithm 1 Benchmark algorithm

1: **if** $mode = 16x16$ **then**
2: Perform ME using λ_i.
3: Store MV^i_{16x16}.
4: **else**
5: Perform ME using λ_{motion}.
6: **end if**
7: Perform MD using $\lambda_i \forall i$.
8: **return** Best mode.
9: **return** λ_i^*.

Table 3.4: Summary of coding conditions.

Parameter	**Value**
GOP	IPPP
fps	30
QP values	[20, 24, 28, 32]
RDO	ON
# RFs	1
# Frames	100
Modes	16x16 only

applied in order to obtain results in a reasonable amount of time by circumventing the evaluation of all possible combinations (λ_i, *mode*), but taking into account that the 16x16 mode is the most likely to be selected [Martínez-Enríquez et al., 2010].

Following this procedure, λ_i^* values resulting from encoding each MB of several standard video sequences using the coding conditions described in Table 3.4 were gathered. In this case, the number of RFs was set to 1, in order to focus the analysis only on the MV selection.

Since the main interest is to determine in which cases the relationship $\lambda_{motion}(\lambda)$ is not accurate enough, the resulting λ_i^* values have been grouped into three classes: lower, equal, or higher than λ_{motion}. The resulting probabilities along with results in terms of coding performance when comparing to the reference encoding are shown in Table 3.5, where:

Table 3.5: Probabilities (%) of selecting a λ_i^* lower, equal, or higher than $\lambda_{motion} = \sqrt{\lambda}$ for a set of standard sequences and coding performance in terms of $\Delta R(\%)$ and $\Delta Y(dB)$.

	$P(\lambda_i^* < \lambda_{motion})$	$P(\lambda_i^* = \lambda_{motion})$	$P(\lambda_i^* > \lambda_{motion})$	$\Delta R(\%)$	$\Delta Y(dB)$
Akiyo	1.31	92.59	6.10	-1.72	0.07
Coastguard	6.89	59.90	33.21	-1.19	0.07
Foreman	6.31	64.64	29.05	-3.37	0.15
Highway	11.17	69.78	19.05	-4.21	0.08
Ice Age	0.36	93.73	5.91	-4.75	0.31
Nature	2.62	86.02	11.36	-2.94	0.14
News	2.08	88.55	9.37	-1.92	0.10

- $P(\lambda_i^* = \lambda_{motion})$ represents the probability of selecting λ_{motion} as the best coding option.
- $P(\lambda_i^* < \lambda_{motion})$ represents the probability of selecting $\lambda_i^* < \lambda_{motion}$, therefore giving more weight to the D_{motion} term.
- $P(\lambda_i^* > \lambda_{motion})$ represents the probability of selecting $\lambda_i^* > \lambda_{motion}$, thus putting more emphasis on the R_{motion} term.

Finally, the last two columns represent the gain in terms of coding performance.

According to the obtained results, choosing λ_{motion} as the optimal one is undoubtedly the most likely. Nevertheless, there is a significant probability of selecting a λ_i^* different from λ_{motion}.

After carrying out a further analysis on the obtained results aiming to find common visual features that explain these results, it has been noted that video sequences presenting size-changing objects (e.g., *zoom*, approaching objects), such as *Highway*, lead to obtain a higher $P(\lambda_i^* < \lambda_{motion})$; in sequences exhibiting high translational movements, such as *Foreman* or *Coastguard*, a higher $P(\lambda_i^* > \lambda_{motion})$ is obtained; and finally, in sequences showing low-motion content, such as *Akiyo* or *Ice Age*, λ_{motion} becomes optimal with high probability.

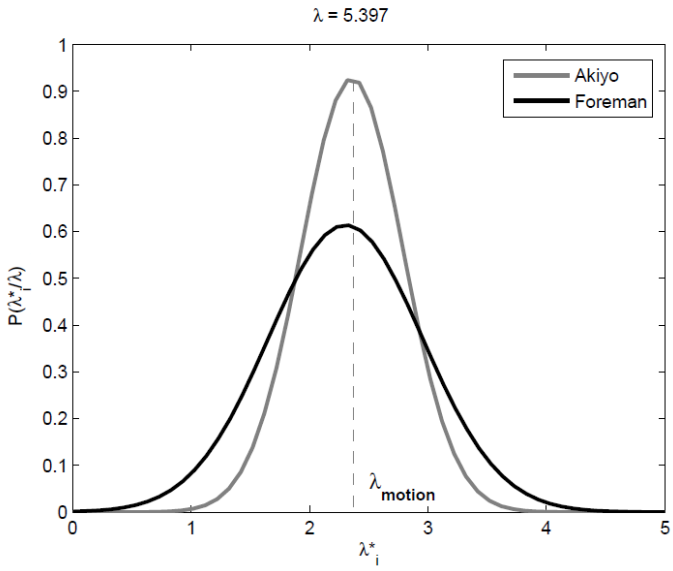


Figure 3.1: Conditional pdf of λ_i^* given $\lambda = 5.397$ ($P(\lambda_i^*/\lambda = 5.397)$). $\lambda = 5.397$ corresponds to $QP = 20$, which is used for high quality encodings.

From the encoding performance results, two conclusions arise. First, it does not appear to be a strong correlation between $P(\lambda_i^* \neq \lambda_{motion})$ and the improvement in coding performance, as *Ice Age* is the most likely to have an optimal $\lambda_i^* = \lambda_{motion}$ and, on the other hand, is the one that shows a higher improvement (this specific example will be further explained later). Second, improvements increase with respect to the ones shown in Table 3.3, where F remained constant along the video sequence, so the adaptation ability seems to have a positive impact on the results.

With the aim of illustrating these ideas with specific examples, Figure 3.1 shows the conditional probability density function (pdf) of λ_i^* given λ, $P(\lambda_i^*/\lambda)$, for *Akiyo* and *Foreman*. As long as the relation $\lambda_i^* = \lambda_{motion}$ is accurate, the mean of $P(\lambda_i^*/\lambda)$ would tend to λ_{motion} and its variance would tend to zero. As can be observed, the variance is higher in *Foreman* than in *Akiyo*, and $P(\lambda_i^* = \lambda_{motion})$ is significantly lower.

These results show some correlation between λ_i^* and motion content and, accord-

ing to Table 3.5, this correlation can be observed for the rest of the video sequences. In particular, λ_{motion} is not an accurate estimation for sequences such as *Foreman*, which exhibits random motion due to the hand-holding camera and the large head movements. In contrast, λ_{motion} turns out to be quite an accurate estimation for sequences such as *Akiyo*, which was captured with a static camera and shows small head movements.

In accordance with these results, which suggest that there seem to be certain correlation between the motion content and the accuracy of the $\lambda_{motion}(\lambda)$ relation, [Zhang et al., 2010] identified content-related events for which λ_{motion} needs to be adjusted to improve $R-D$ performance. In particular, motion content described by high module and random-pointing MVs will be better coded by means of a modified version of the $\lambda_{motion}(\lambda)$ relationship.

This malfunction has been related in the literature with the *high rate* model implemented in the JM15.1 for the R, as only the transformed coefficients are considered in the model. Therefore, whenever the side information (MVs, headers) is comparatively similar to the transformed coefficients information, which is whenever the block-matching model fails to produce small differential MVs, the model tends to be inaccurate, and this can be compensated by means of increasing λ_{motion}. Some video transitions that compromise the block-matching model are described in [Boyce, 2004], [Budagavi, 2005] and [Kamikura et al., 1998], naming the complex translational movements, *rotations*, *fades* or *blurring*.

To prove this hypothesis, the previous analysis is repeated focusing on selected video segments for which it is known *a priori* that ME does not work correctly, such as non-translational events (*fade* transitions, *rotation*, *blurring*, etc.) or complex movements.

To this purpose, λ_i^* values resulting from encoding each MB of a set of selected

Table 3.6: Probabilities (%) of selecting a λ_i^* lower, equal, or higher than $\lambda_{motion} = \sqrt{\lambda}$ for a selected set of ME-compromising video segments.

	# Frames	$P(\lambda_i^* < \lambda_{motion})$	$P(\lambda_i^* = \lambda_{motion})$	$P(\lambda_i^* > \lambda_{motion})$	$\Delta R(\%)$	$\Delta Y(dB)$
Airshow (rotation)	150	3.76	70.52	25.72	-6.01	0.34
Corvette (fade in)	8	5.74	36.05	58.21	-14.19	0.77
Ice age (cross-fade)	42	4.36	55.58	40.06	-8.89	0.60
Sintel (rapid mov.)	73	3.40	73.98	22.62	-5.61	0.37

video segments (using the same encoder configuration as in Table 3.4) are gathered. Results are shown in Table 3.6. When comparing these results to those of Table 3.5, which referred to standard sequences, it becomes obvious that the probability of $\lambda_i^* = \lambda_{motion}$ is significantly lower for these selected ME-compromising segments. Furthermore, for the particular case of *fade* transitions, $P(\lambda_i^* > \lambda_{motion})$ is comparatively as high as $P(\lambda_i^* = \lambda_{motion})$ or even higher for the case of *Corvette*. This result is due to that ME is not properly managing the illumination changes and this fact affects the whole frame in such a manner that every MB in the frame is affected by this inaccurate ME.

This also explains the results shown for *Ice Age* in Table 3.5. As *Ice Age* is a video sequence which shows *fade* transitions between near-static video fragments, most of the time is showing static video content similar to *Akiyo* and using the reference λ_{motion} value. However, on frames in which those transitions occur $\lambda^* \neq \lambda_{motion}$ is selected, affecting the whole frame and subsequently, obtaining important improvements in terms of coding performance.

Another aspect to take into account is that, apart from the weaknesses of the *high rate* model mentioned in [Zhang et al., 2010], there are also situations in which the λ_{motion} multiplier needs to be diminished to increase the weight of the D_{motion} term. Therefore, an additional conclusion can be extracted from Table 3.6. The improvement in coding performance is significantly higher than the one obtained

with the previous set of video sequences, and part of this improvement may be due to the use of $\lambda^* < \lambda_{motion}$ in some frames.

In summary, it is hypothesized that the estimation of the Lagrangian parameter in J_{motion} can be improved for ME-compromising events. In other words, although every sequence shows a certain percentage of $\lambda_i^* \neq \lambda_{motion}$, it is specially in these cases where the estimation of the Lagrangian parameter in J_{motion} should be adapted to produce a MV more similar to the one that would be obtained by evaluating J.

3.1.3 J_{motion} as a low-complexity alternative to J

In this section the differences between J_{motion} and J are discussed in order to gain insight into the causes that may lead to poor performance of J_{motion}.

To find the optimal MV, the ME process should ideally evaluate J for all the points in the search area. Given that this process is not computationally feasible, the ME process optimizes J_{motion} instead (2.13), which can be viewed as a low-complexity estimation of J and can be rewritten (from (2.13)) as follows:

$$J_{motion} = \sum_{(x,y)\in MB} \left| I\left(x,y\right) - \widehat{I}\left(x,y\right) \right| + \lambda_{motion} R_{motion}, \tag{3.5}$$

where x and y are the horizontal and vertical coordinates within the MB; $I\left(x,y\right)$ is the luminance of the pixel (x,y) in the original MB; $\widehat{I}\left(x,y\right)$ is the luminance of the pixel (x,y) in the predicted MB; and R_{motion} is an estimation of the amount of bits needed to encode the residual transformed coefficients. In other words, the goal of the ME process is to obtain, by minimizing J_{motion}, the same (MV, RF) pair that

would have been obtained by optimizing J, which can be rewritten as follows:

$$J = \sum_{(x,y)\in MB} \left(I\left(x,y\right) - \widetilde{I}\left(x,y\right) \right)^2 + \lambda \left(R_{coeffs} + R_{side} \right), \tag{3.6}$$

where $\widetilde{I}\left(x,y\right)$ is the luminance of the pixel (x,y) of the reconstructed MB; R_{coeff} is the amount of bits allocated to the transformed coefficients information; and R_{side} represents the side information needed to represent the MV, RF, mode, headers, etc.

The difference between the distortion terms in (3.5) and (3.6) comes from the SAD calculation and the use of the predicted reference $\widehat{I}\left(x,y\right)$ in J_{motion} instead of the SSD calculation and the reconstructed MB $\widetilde{I}\left(x,y\right)$ in J. The difference between the rate terms is also clear: J_{motion} uses an estimation of the bits allocated to the reconstructed coefficients, while J considers the actual rate including also the side information.

Thus, J_{motion} relies on low-complexity estimations of the R and D terms in J. When these estimations produce significantly different errors, the balance between R_{motion} and D_{motion} moves from that of D and R, making the minimization of J_{motion} to, very likely, fail to produce the same MV than that of J. In these cases, one option could be to adapt λ_{motion} to compensate for this unbalance.

3.1.4 When J_{motion} does not work properly: an illustrative example

To illustrate the correlation between the ME-compromising situations and the lack of accuracy of the $\lambda_{motion}(\lambda)$ relation, in this section an example that deals with a *cross-fade* transition is developed. *Fade* transitions are characterized by general illumination changes that severely affect the performance of the block-matching model

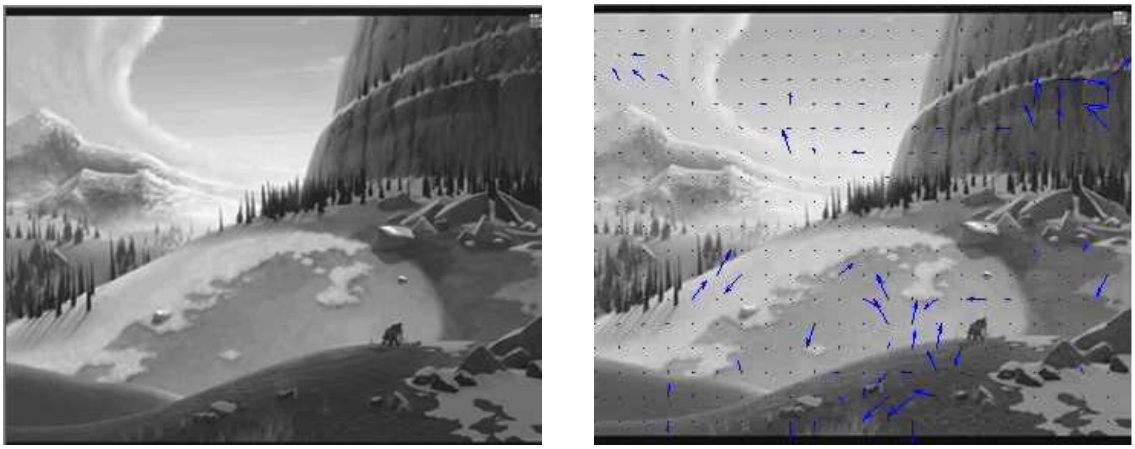
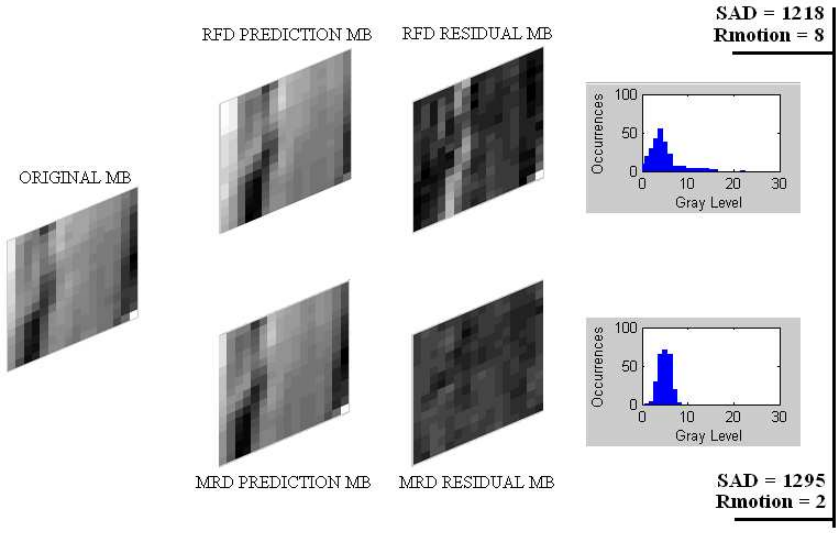


Figure 3.2: Frames #253 (left) and #254 (right) of *Ice Age*. MVs are superimposed on the #254 frame.

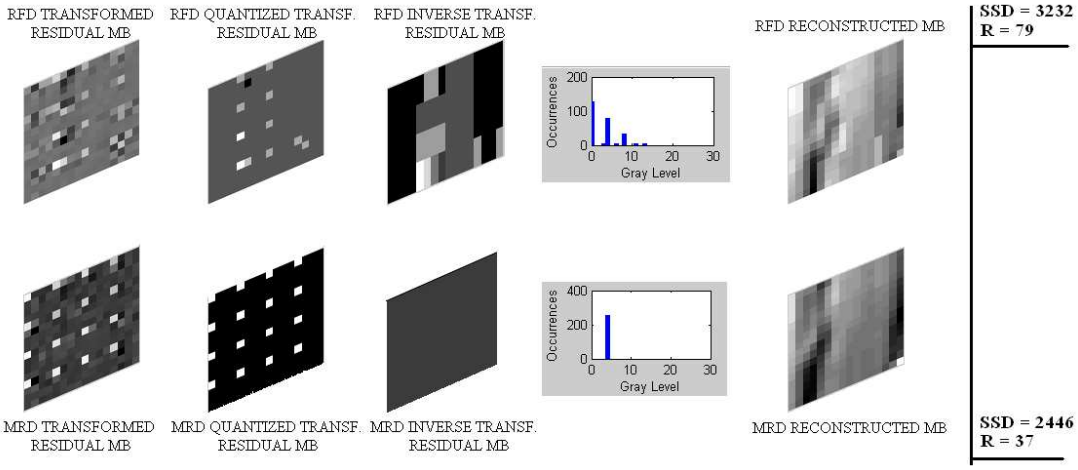
implemented in the reference software JM15.1, which was specifically designed for translational movements and is not able to cope with illumination changes[2].

The selected example consists of a *cross-fade* happening between two consecutive frames (#253 and #254) of *Ice Age*. Figure 3.2 shows the two considered frames, where it can be seen that frame #254 is comparatively lighter than frame #253, due to the transition. In this example, first, the reference software implementation is used and the MV is selected by optimizing J_{motion}. This approach will be referred as Reference Decision (RFD). The MVs obtained following the RFD approach (using the frame #253 as reference) are superimposed on frame #254. As can be observed, some large MVs appear on regions where there is no actual movement. These MVs appear due to illumination changes that make the ME find in the search area positions with similar mean luminance comparing with the original MB, minimizing the SAD sufficiently to be worth sending a MV. However, intuitively, the co-located MB seems to be the best option, as it would not need to send a MV and only the DCT coefficients

[2]It should be noted that there are specific methods to deal with illumination changes, such as weighted prediction [Boyce, 2004], but such solutions are out of the scope of this work since they are not centered around the RDO process.



(a) ME process



(b) MD process

Figure 3.3: Comparative illustration of RFD (top row) and MRD (bottom row) in both the ME (a) and MD (b) processes.

would be needed to encode the residue.

In order to explain this hypothesis in more detail, a different ME process which uses a modified cost function $\widehat{J}_{motion}$ (3.4) was employed, allowing us to deliberately alter the balance between D_{motion} and R_{motion}. An arbitrarily large F_i value is employed in order to select the MV that minimizes R_{motion}, allowing to evaluate the

co-located MB as a coding option. This approach will be called Minimum Rate Decision (MRD).

Figure 3.3 shows a parallel analysis of the two processes considered, RFD (top) and MRD (bottom) in order to prove our hypothesis. Figure 3.3(a) focuses on the ME process (J_{motion} optimization) showing the predicted MBs, the residues, and their histograms. Figure 3.3(b) focuses on the MD process (J optimization) depicting the DCT coefficients (before and after the quantization process), the reconstructed residues, along with their histograms and the reconstructed MBs.

As can be observed in Figure 3.3(a), the RFD residue presents a lower mean value than that of MRD, which ultimately leads to a lower SAD. Moreover, the difference in SAD values is high enough to be worth sending MV information, as the R_{motion} for RFD is higher. However, it should be noted that the variance of the residual is higher for the RFD residue than for that of MRD. Considering that MRD points to the co-located MB and the transition is just an illumination change, this higher variance makes us think that more AC coefficients will be needed to encode the RFD proposal, compared with the MRD one.

Nonetheless, at the end of the ME process in Figure 3.3(a), the JM15.1 reference software would select RFD as best.

Moving forward to Figure 3.3(b), when the DCT coefficients are obtained, it becomes clear that the RFD transformed residual presents higher AC coefficients and, on the contrary, the MRD transformed residual mainly presents DC coefficients (changes in mean illumination), as it was hypothesized. Therefore, when reconstructing the MBs, the illumination change is being properly modeled by sending the DC coefficients only, and the SSD value is significantly lower in comparison with that of RFD, which had more information in the quantized AC coefficients.

Furthermore, the cost in terms of R for RFD is higher because of the MV that

needs to be sent besides the AC coefficients, which are less efficiently coded than the DC ones because they do not provide long runs to the entropy coding phase. However, the R_{motion} estimation seems to have been more accurate than the D_{motion} one, as it was stated in the ME process that the RFD solution would be more expensive in terms of allocated bits.

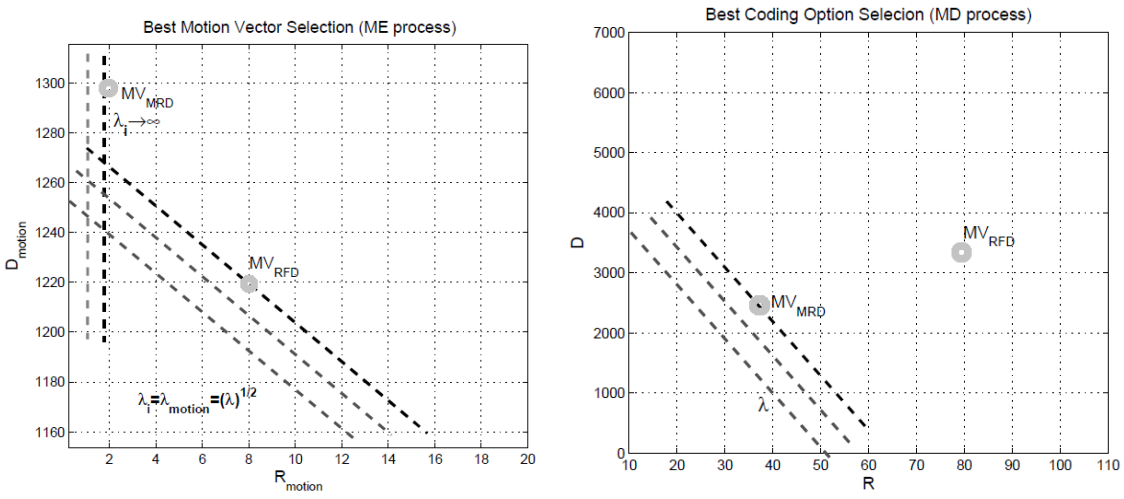
As a result, MRD provides a more efficient solution to the optimization problem than RFD, and this has been caused by an overestimated D_{motion} value in MRD, making the balance between D_{motion} and R_{motion} to be dominated by the estimation error in D_{motion}: although R_{motion} for RFD is significantly higher than that of MRD, the MV chosen is not the co-located because of the SAD term.

Figure 3.4 provides a graphical explanation from the Lagrange optimization theory point of view [Ortega and Ramchandran, 1998], using real $R - D$ data taken from the previous example.

The ME process is illustrated in the left part of the figure, where the two compared solutions are depicted in the $R_{motion} - D_{motion}$ space labeled as MV_{RFD} (MV associated with RFD) and MV_{MRD} (MV associated with MRD). The optimal solution for a given λ_i is the operating point in the $R_{motion} - D_{motion}$ space that first hits a plane wave of slope λ_i (dashed lines in Figure 3.4). Therefore, as shown, MV_{RFD} becomes the best solution for $\lambda_i = \lambda_{motion}$ while MV_{MRD} provides the lowest-rate solution (λ_i arbitrarily large).

The MD process is shown in the right part of the figure. The two compared operating points, MV_{RFD} and MV_{MRD}, are depicted in the $R - D$ space and the optimal solution for a given λ is the one that is first overlapped by a plane wave of slope λ. In this case, where the terms R and D are not estimations, MV_{MRD} becomes the optimal solution for this particular example.

It can also be seen from the MV_{RFD} and MV_{MRD} positions in both cases that



(a) Coding option selection in the ME process. (b) Coding option selection in the MD process.

Figure 3.4: Graphical illustration of the optimal coding option selection.

an important error in the estimation of the D term has been made, as their relative positions in terms of R are similar, but in terms of D, their positions significantly change from ME to MD.

In the same manner as has been explained so far, it is natural to think of the inverse situation for the cases in which $\lambda_i^* < \lambda_{motion}$: sometimes λ_{motion} would produce a solution for which the R_{motion} term is overestimated (R_{motion}-biased solution) that could be corrected by giving more weight to the D_{motion} term (D_{motion}-driven solution), which can be implemented just by using $F_i = 0$. This alternative approach will be referred to as Minimum Distortion Decision (MDD).

In summary, ME-compromising situations can lead to R_{motion} or D_{motion}-biased solutions, which will require the encoder to be able to select a different λ_{motion} value in order to make a more accurate decision in J_{motion}. From this study, the inefficient *high rate* approximation considered in [Zhang et al., 2010] has been generalized, by also characterizing situations in which the D_{motion} term needs to be strengthened. It is also important to note that these biased solutions can occur in every MB of

every video sequence. However, as it has been studied, the likelihood of occurrence is significantly higher in these ME-compromising situations.

3.2 Proposed Method

This Section describes a computationally efficient method to find a more suitable value of λ_{motion}. In previous experiments, 21 different F_i values were evaluated for each MB, which is computationally unfeasible. Therefore, an statistical analysis on which F_i values are more likely to be selected will be carried out first. Then, a new method will be proposed that applies extra F_i evaluations only in the cases in which a change in the reference $\lambda_{motion}(\lambda)$ relationship likely improves the coding efficiency.

3.2.1 Reduced set of λ_i values

The modified cost function $\widehat{J}_{motion}$ involving 21 different factors F_i has been useful to set the motivation for this work, but becomes computationally impractical for coding purposes. Therefore, it is necessary to propose an alternative that allows us to take advantage of using a more suitable Lagrangian parameter in J_{motion} without incurring a significant increase of the computational cost.

To this end, we decided to select a reduced set of three λ_i values: one higher than λ_{motion}, which would allow for compensating D_{motion}-biased solutions, one lower than λ_{motion}, which would allow for compensating R_{motion}-biased solutions, and λ_{motion}. In so doing, it seems reasonable to select the extremes, $\lambda_i = 0$ and λ_i arbitrarily large (hereafter called $\lambda_i \to \infty$), since they would allow for avoiding the potentially largest errors. Interestingly, $\lambda_i = 0$ corresponds to the MDD discussed previously, and $\lambda_i \to \infty$ to the MRD.

Table 3.7: Probability (%) of selecting each λ_i value.

i	$P(\lambda_i^* = \lambda_i)$	i	$P(\lambda_i^* = \lambda_i)$
0	1.94	11	1.01
1	0.51	12	0.82
2	0.60	13	0.68
3	0.45	14	0.64
4	0.37	15	0.50
5	82.58	16	0.47
6	2.27	17	0.39
7	1.96	18	0.39
8	1.67	19	0.33
9	1.27	20	0.33
10	1.16		

To study the suggested solution more in depth, λ_i values from encoding each MB of a set of video segments were gathered.

For these experiments, we used an IPPP GOP pattern at 30 fps, four QP values (20, 24, 28, 32) and RDO enabled (both video segments and encoder configuration are further described in Section 3.3). The obtained results are shown in Table 3.7, where the reference value, λ_{motion}, is labeled as $i = 5$, which corresponds to $F_5 = 0.2 \times 5 = 1$.

Regarding the $\lambda_i < \lambda_{motion}$ values, it seems reasonable to select $\lambda_i = 0$ since it clearly exhibits the highest probability among the λ_i values which increase the influence of the D_{motion} in the cost function.

Regarding the $\lambda_i > \lambda_{motion}$ values, the probabilities are dispersed and they decrease with the increment of i, which suggests that taking a unique $\lambda_i > \lambda_{motion}$ value is unsuitable. However, it has been observed that whenever a certain λ_i reaches the global minimum of R_{motion} by selecting the predicted motion vector (MV_p), any λ_j (with $j > i$) will obtain the same pair of (MV, RF), as more emphasis is applied on the R_{motion} term, which has already reached the global minimum, and the cost function will provide the same result ($J_{motion}(\lambda_i) = J_{motion}(\lambda_j)$). Moreover, this behavior

will lead to choose λ_i as best in our study[3].

Thus, taking this behavior into consideration, it was studied the probability of selecting as optimal the λ_i that first yields the MV_p as a solution for the cost function minimization, obtaining a 69% of selection among all the other cases. Then, it is proposed to compensate the D_{motion}-biased solutions with the λ_i value that leads to the MV_p as optimal, which ultimately can be represented as $\lambda_i \to \infty$.

As a conclusion, $\lambda_i = 0$ and $\lambda_i \to \infty$ (MDD and MRD, respectively) are proposed as statistical good candidates to be evaluated in the ME process of each MB, also fulfilling the goal of avoiding large errors.

In terms of computational efficiency, it must be highlighted that during the ME process, the D_{motion} and R_{motion} terms are computed for each position in the search area. Therefore, only one ME pass is required to obtain the three MVs sought. Subsequently, these MVs should be tested on J to obtain the optimal coding option.

To reduce the computational cost associated with the two additional J evaluations, it is proposed to assess MDD and MRD only for the 16x16 pixel block size, which is the most likely one [Martínez-Enríquez et al., 2010] and assess the rest of coding modes only with the λ_{motion} value. Furthermore, to achieve higher computational savings, when the reference pair (MV, RF) turns out to be the same than that obtained by either MDD or MRD, only this reference pair (MV, RF) is tested in the MD process since the third option becomes very unlikely (MDD and MRD actually represent "opposite" solutions). As a result, as empirically shown in the next section, the proposed coding process does not incur a significant increment of the computational cost with respect to the reference coding process.

[3]The implementation has been done in a way that whenever two different λ_i produce an equal J in the MD stage, the lower λ_i is selected as best.

3.2.2 Summary of the Algorithm

The complete algorithm is summarized in Algorithm 2.

Algorithm 2 Proposed coding process of an MB.

1: {**ME process** (J_{motion})}
2: Obtain (MV_{RFD}, RF_{RFD}), (MV_{MRD}, RF_{MRD}), and (MV_{MDD}, RF_{MDD}) for 16x16 block size.
3: Obtain (MV_{RFD}, RF_{RFD}) for the remaining available modes.
4: {**MD process** (J)}
5: **if** $(MV_{RFD}, RF_{RFD}) \neq (MV_{MRD}, RF_{MRD})$ and $(MV_{RFD}, RF_{RFD}) \neq (MV_{MDD}, RF_{MDD})$. **then**
6: Test (MV_{RFD}, RF_{MRD}), (MV_{MRD}, RD_{MRD}), and (MV_{MDD}, RF_{MDD}) for 16x16 on J.
7: Test (MV_{RFD}, RF_{RFD}) for the remaining available modes on J.
8: **else**
9: Test (MV_{RFD}, RF_{RFD}) for all the available modes on J.
10: **end if**
11: Select optimal mode: $\min J$.
12: **return** Best mode.

3.3 Experimentation

The proposed algorithm was implemented in the H.264/AVC JM15.1 reference software [JVT, 2010]. The test conditions were selected according to the recommendations of the JVT [Sullivan, 2001], namely: main profile, ±32 pixel search range, Context-Adaptive Binary Arithmetic Coding (CABAC), and RDO enabled. Moreover, an IPPP GOP pattern and four QP values (20, 24, 28 and 32) were used. Table 3.8 summarizes the encoder configuration.

To assess the proposed algorithm in terms of $R-D$ performance, the average bitrate differences ($\Delta R(\%)$) and the average PSNR differences ($\Delta Y(\text{dB})$) of the luma component were used, as in Section 3.1.1. Moreover, to evaluate the computational complexity of the proposed algorithm, the time increment ($\Delta T(\%)$) was calculated as follows:

$$\Delta T = \frac{T_{method} - T_{JM15.1}}{T_{JM15.1}} \times 100(\%), \tag{3.7}$$

where T_{method} is the encoding time of the proposed method and $T_{JM15.1}$ is the encoding time of the reference JM15.1 software.

The proposed algorithm was tested with respect to the H.264/AVC reference software and with respect to an state-of-the-art algorithm called Context-Adaptive Lagrange Multiplier (CALM) [Zhang et al., 2010], which suggests a context adaptive adjustment of λ_{motion} based on thresholds to improve coding efficiency. The comparative assessment was performed on a varied set of video segments exhibiting ME-compromising events to show the improved performance of the proposed algorithm in these cases. This video segments are of three different resolutions: CIF 352x288, Standard Definition (SD) 720x576 and High Definition (HD) 1280x720.

Since the proposed algorithm aims to improve the ME process, it was first tested avoiding potential interference from spatial prediction tools (Intra modes in Inter

Table 3.8: Encoder configuration.

Parameter	**Value**
Profile IDC	Main
QP	20, 24, 28, 32
GOP	IPPP @ 30 fps.
ME algorithm	Fast Full Search
Search Range	±32
# RFs	3
Symbol Mode	CABAC
RDO	ON

frames were disabled). Then, the coding performance was tested adding the spatial prediction tools (this will be referred to as overall coding performance). Additionally, an upper performance bound was computed resulting from assessing a large set of λ_i values.

Subsequently, the contribution of each part of the proposed algorithm (MDD and MRD) was analyzed in detail. Finally, two illustrative examples of the improved subjective quality achieved by the proposed algorithm are also provided.

3.3.1 Evaluation of the ME performance

The proposed method aims to improve the performance of the ME process by avoiding suboptimal choices of MVs. Therefore, the first experimental evaluation was directed to assess the actual improvement of the ME performance. To this end, the use of *Intra modes in Inter Frames* was disabled since this coding tool can mask failures of the block-matching model. Table 3.9 shows the obtained results. For each of the considered sequences, the mean values of $\Delta T(\%)$, $\Delta R(\%)$, and $\Delta Y(\text{dB})$ across the four QP values are shown. Additionally, the last row of the table shows the mean values for all the sequences.

These results reveal that the proposed algorithm clearly improves the JM15.1

coding performance under the same experimental setup. Specifically, the proposed algorithm obtains an average ΔR reduction of -9.27% for the same coding quality with respect to the reference software. Alternatively, these improvements can be seen in terms of ΔY, where the proposed algorithm achieves an average gain of 0.52 dB.

It is important to highlight that a higher gain is obtained on video segments where *fade* transitions take place, such as *Mobisode* or *Corvette*, where ΔR reductions of -21.18% and -32.60% are obtained, respectively (0.82 dB and 1.95 dB in terms of ΔY). This is due to the fact that the optimal value of λ_{motion} in these cases is different from the reference one with high probability, as it was shown in Section 3.1.2 for *Corvette* in Table 3.6.

Comparing with CALM algorithm, the proposed method produces better coding quality with a slightly higher complexity increment. It should be noted, however, that CALM works better in the low-complexity RDO scenario (RDO off).

Regarding the computational complexity, a good compromise has been achieved as ΔT reaches an average value of 3.07% comparing with the reference software and 1.74% comparing with CALM, while providing very significant improvements in terms of $R - D$ performance. Moreover, looking at the individual video segments, the highest value of ΔT incurred by the proposed method is close to 4%, while the worst case for CALM is close to 14%.

3.3.2 Evaluation of the overall coding performance

To evaluate the overall coding performance, the *Intra modes in Inter Frames* coding option was enabled in the JM15.1 reference software. It is expected that the use of the Intra mode coding tool compensates for some ME failures and, consequently, the performance improvement achieved by the proposed algorithm is lower than the one

Table 3.9: Performance evaluation of the proposed algorithm relative to JM15.1 with Intra coding in Inter frames disabled. Comparative results of CALM [Zhang et al., 2010] are also provided.

			Proposed method			CALM		
Sequence	**Effect**	**# Frames**	$\Delta T(\%)$	$\Delta R(\%)$	ΔY (dB)	$\Delta T(\%)$	$\Delta R(\%)$	ΔY (dB)
Ice Age (CIF)	*cross-fade*	42	1.87	−9.34	0.64	0.16	0.40	0.02
Ice Age (CIF)	*cross-fade*	13	2.89	−11.24	0.76	0.19	0.50	0.02
Nature (CIF)	*blurring*	100	2.08	−1.67	0.08	0.00	0.01	0.00
Airshow (SD)	*rotation*	150	3.33	−6.64	0.40	0.20	0.56	0.00
Corvette (SD)	*fade in*	8	2.96	−32.60	1.95	0.63	−0.75	0.06
Corvette (SD)	*zoom in*	50	4.03	−0.40	0.01	1.68	−0.01	0.00
Corvette (SD)	*zoom out*	5	2.73	−0.58	0.03	14.06	−0.10	0.01
Mobisode (SD)	*cross-fade*	20	2.34	−21.18	0.82	−1.15	0.84	0.01
Controlled Burn (HD)	*cross-fade*	10	3.16	−15.86	0.82	−0.38	−1.77	0.10
Dinner (HD)	*blurring*	62	3.98	−4.22	0.22	0.11	−0.73	0.02
Dinner (HD)	*zoom out*	100	4.10	−1.05	0.05	−0.19	−0.24	0.01
Sintel (HD)	*rapid mov.*	73	3.37	−6.43	0.41	0.64	−0.21	0.01
Average			**3.07**	**−9.27**	**0.52**	**1.33**	**−0.13**	**0.02**

obtained in section 3.3.1.

The obtained results are shown in Table 3.10, where an average −2.20% of ΔR reduction is achieved in comparison with the reference software. Alternatively, in ΔY terms, an improvement of 0.12 dB is obtained. On the one hand, the best results continue to appear in sequences exhibiting *fade* transitions such as *Ice Age* and *Corvette* for the same reasons (now softened by the use of the Intra modes). On the other, the performance improvements becomes less relevant in *zoom*-type transitions, where the results tend to be similar to those of the reference.

In summary, it can be concluded that despite the use of the Intra mode coding tool overcomes some of the problems associated with ME-compromised events, the proposed algorithm still provides significant $R - D$ improvements in exchange for a low increment of computational complexity. Moreover, the evaluation of the reference λ_{motion} prevents the proposal from incurring significant losses when the alternative value does not apply.

As can be seen for the experimental protocol used in this paper, CALM does

Table 3.10: Performance evaluation of the proposed algorithm relative to JM15.1 with Intra coding in Inter frames enabled. Comparative results of CALM [Zhang et al., 2010] are also provided.

			Proposed method			CALM		
Sequence	**Effect**	**# Frames**	$\Delta T(\%)$	$\Delta R(\%)$	ΔY (dB)	$\Delta T(\%)$	$\Delta R(\%)$	ΔY (dB)
Ice Age (CIF)	*cross-fade*	42	−0.55	−7.57	0.50	−0.65	−0.47	0.02
Ice Age (CIF)	*cross-fade*	13	1.21	−4.98	0.32	−1.25	−0.24	0.02
Nature (CIF)	*blurring*	100	1.84	−1.81	0.09	0.51	0.26	−0.01
Airshow (SD)	*rotation*	150	3.22	−0.85	0.04	0.47	−0.03	0.01
Corvette (SD)	*fade in*	8	5.17	−6.21	0.28	1.24	−0.15	0.01
Corvette (SD)	*zoom in*	50	3.79	−0.14	0.00	−0.30	0.00	−0.01
Corvette (SD)	*zoom out*	5	5.12	−0.56	0.03	2.21	−0.04	0.00
Mobisode (SD)	*cross-fade*	20	3.53	−2.70	0.06	−0.54	0.76	−0.03
Controlled Burn (HD)	*cross-fade*	10	2.29	−1.18	0.04	−0.88	0.03	0.00
Dinner (HD)	*blurring*	62	3.33	0.19	0.00	0.07	0.04	0.00
Dinner (HD)	*zoom out*	100	3.55	−0.30	0.01	0.75	0.00	0.00
Sintel (HD)	*rapid mov.*	73	3.32	−0.31	0.02	1.23	−0.13	0.00
Average			**2.99**	**−2.20**	**0.12**	**0.24**	**0.00**	**0.00**

not provide any average improvement with respect to the reference software, likely because it was conceived for RDO-disabled operation.

Finally, note that the computational cost is just slightly higher in the proposed algorithm than in the reference software. Specifically, using the proposed algorithm implies a 2.99% increment of ΔT with respect to the reference.

3.3.3 An upper performance bound

An extended version of the algorithm that assesses 40 different λ_{motion}values was also tested with the aim of providing an upper performance bound. The procedure described in Section 3.1.2 was used for $i \in [0, 1, \cdots, 40]$ in (3.3). Table 3.11 shows comparative results between the proposed method and this upper performance bound.

As can be observed, although the upper performance bound clearly improves the results of the proposed method, the room for improvement is quite moderate in average. However, when considering some specific video sequences as *Ice Age* or *Corvette (fade in)*, the upper performance bound is not better in terms of coding efficiency,

Table 3.11: Performance evaluation of the proposed algorithm with respect to an empirical upper bound. Results in both cases are relative to JM15.1 with Intra coding in Inter frames enabled.

			Proposed method			Upper Bound		
Sequence	**Effect**	**# Frames**	$\Delta T(\%)$	$\Delta R(\%)$	ΔY (dB)	$\Delta T(\%)$	$\Delta R(\%)$	ΔY (dB)
Ice Age (CIF)	*cross-fade*	42	−0.55	−7.57	0.50	1924	−6.79	0.48
Ice Age (CIF)	*cross-fade*	13	1.21	−4.98	0.32	2022	−4.42	0.29
Nature (CIF)	*blurring*	100	1.84	−1.81	0.09	1748	−2.58	0.12
Airshow (SD)	*rotation*	150	3.22	−0.85	0.04	1927	−2.34	0.11
Corvette (SD)	*fade in*	8	5.17	−6.21	0.28	2130	−5.70	0.24
Corvette (SD)	*zoom in*	50	3.79	−0.14	0.00	1977	−2.11	0.09
Corvette (SD)	*zoom out*	5	5.12	−0.56	0.03	1908	−2.26	0.13
Mobisode (SD)	*cross-fade*	20	3.53	−2.70	0.06	2106	−4.61	0.14
Controlled Burn (HD)	*cross-fade*	10	2.29	−1.18	0.04	2089	−1.81	0.06
Dinner (HD)	*blurring*	62	3.33	0.19	0.00	1984	−1.41	0.06
Dinner (HD)	*zoom out*	100	3.55	−0.30	0.01	2064	−3.49	0.15
Sintel (HD)	*rapid mov.*	73	3.32	−0.31	0.02	2036	−0.97	0.05
Average			**2.99**	**−2.20**	**0.12**	**1993**	**−3.21**	**0.16**

compared with the proposal (as can be expected). The reason is that decisions made are locally optimal, following the independence consideration between MBs (seen on Section 2.2.2), but since they actually affect the encoding of the neighboring MBs, sometimes they can be globally sub-optimal. This is an empirical example of the actual inter-dependency existing between decisions in different MBs.

Nonetheless, these results allow us to conclude that, although there is some room for improvement, the proposed solution provides an excellent balance between performance and computational cost: it achieves −2.20% bit-rate reduction (0.12 dB) vs. −3.21% of the upper bound (0.16 dB) without incurring a significant increment of the computational cost.

3.3.4 Evaluation of the MRD and MDD contributions

An analysis of the individual contributions of both MRD and MDD was performed to assess their relative influence on the global performance. Table 3.12 shows the overall coding performance of both MRD and MDD with respect to the reference software.

Table 3.12: Independent performance evaluation of MRD and MDD.

			MRD			MDD		
Sequence	**Effect**	**# Frames**	$\Delta T(\%)$	$\Delta R(\%)$	ΔY (dB)	$\Delta T(\%)$	$\Delta R(\%)$	ΔY (dB)
Ice Age (CIF)	*cross-fade*	42	−0.46	−6.97	0.48	3.00	−7.14	0.49
Ice Age (CIF)	*cross-fade*	13	−0.04	−4.00	0.26	2.88	−4.12	0.27
Nature (CIF)	*blurring*	100	0.81	−1.86	0.09	4.28	−2.16	0.09
Airshow (SD)	*rotation*	150	2.07	−0.23	0.01	3.92	−0.28	0.01
Corvette (SD)	*fade in*	8	3.04	−6.61	0.29	4.15	−6.49	0.29
Corvette (SD)	*zoom in*	50	3.36	0.52	−0.03	3.85	0.46	−0.02
Corvette (SD)	*zoom out*	5	1.75	0.15	0.00	3.50	0.17	0.00
Mobisode (SD)	*cross-fade*	20	−1.42	−2.89	0.07	1.18	−2.96	0.08
Controlled Burn (HD)	*cross-fade*	10	−0.97	−1.15	0.04	0.39	−1.28	0.04
Dinner (HD)	*blurring*	62	0.66	0.17	0.00	0.32	−0.02	0.01
Dinner (HD)	*zoom out*	100	2.00	0.50	−0.03	1.44	0.48	−0.02
Sintel (HD)	*rapid mov.*	73	3.01	−0.24	0.01	6.33	−0.16	0.01
Average			**1.15**	**−1.88**	**0.10**	**2.74**	**−1.96**	**0.10**

In the first case only MV_{MRD} is considered together with MV_{RFD}. In the second, it is MV_{MDD} the only additional MV considered. Interestingly, it is worth mentioning that the $\Delta T(\%)$ generated by MRD is low in comparison to that of MDD, due to the fact that the probability of MV_{RFD} being the same than MV_{MRD} is higher than for MV_{MDD}.

It is also interesting to notice that, in some particular cases, working just with MRD or MDD outperforms the complete algorithm. The reason is that decisions made are locally optimal (for the current MB), but since they affect the encoding of neighboring MBs sometimes they can be globally sub-optimal (as in the case of the upper-bound). Under general considerations, it can be seen that the use of both decisions, added to the RFD, contribute in a similar manner to obtain better coding performance results when compared with the reference model.

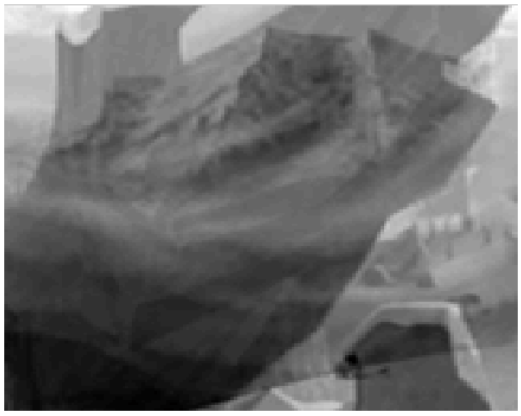
3.3.5 Subjective quality evaluation

Additionally to the objective $R-D$ results shown in previous subsections, a subjective quality analysis of our proposal performance is carried out. To this purpose, two

examples of reconstructed frames from two different video segments, obtained with the reference software JM15.1 and the proposed method, are shown in Figures 3.5 and 3.6.

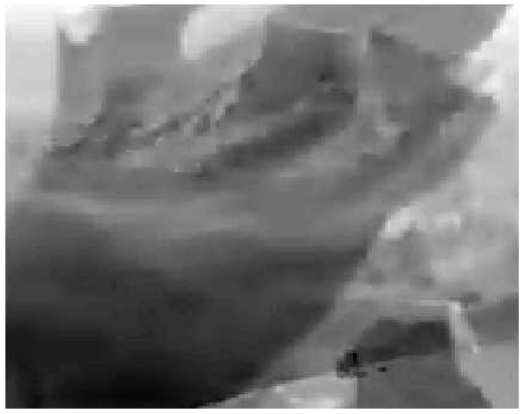
In the first example, one selected frame of the *Ice Age* video segment (specifically, frame # 20) is encoded using both the reference software and our proposal, and the corresponding reconstructions of that frame are comparatively shown. To make this comparison as fair as possible, the QP value was adjusted so that the number of bits produced by this frame would be almost the same in both cases; in particular, it takes up to 8.3 Kb when encoded by the reference software and 8.2 Kb by the proposed method. Figure 3.5 shows three versions of a selected area of the mentioned frame in the *Ice Age* video segment: (a) original; (b) reconstructed by the reference software; and (c) reconstructed by the proposed method. As can be inferred when comparing Figures 3.5(b) and 3.5(c), a higher subjective quality is achieved by the proposed method in comparison with the reference software. Specifically, when looking carefully at the region showing the snowy peak of the mountain, a lot of details are lost in the frame reconstructed by the reference software, while several of them are preserved in the version reconstructed by the proposed method. Another example can be found in the low part of the figures, where two characters (at small size) can be observed: in the reconstructed frame by the reference software one of this characters is missing, while it still appears in the frame reconstructed by the proposed method.

In the second example, one selected frame of the *Mobisode* video segment (specifically, frame # 17) is used. Again, the QP value was adjusted to obtain almost the same number of bits with the reference software and with the described proposal; specifically, 52.5 and 51.1 Kb, respectively. In Figure 3.6 three versions of a selected area are shown (original (a), reconstructed by the reference software (b), and recon-



(a)

(b)

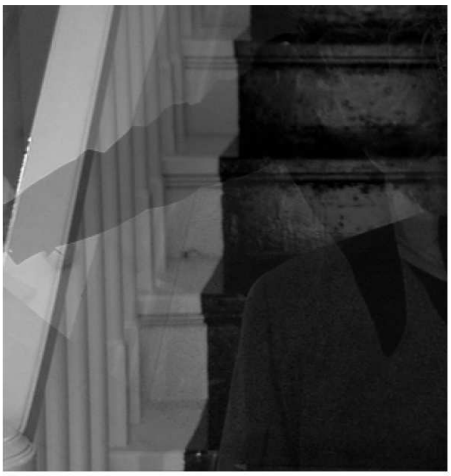


(c)

Figure 3.5: Illustrative example of the achieved subjective quality. (a) Selected part of the original frame #20 of *Ice Age*; (b) reconstructed frame with the reference software; and (c) reconstructed frame with the proposed method.

structed by the proposed method (c)). As it can be observed in Figures 3.6(b) and 3.6(c), in the region showing the bars of the stairs the proposed method achieves better defined edges than the reference software. Moreover, this improvement can be also observed in the shaded peak of the suit in the right part of each figure.

This higher subjective quality can be explained by the fact that the proposed method improves the ME process, encoding some MBs in a more suitable manner. In particular, the ME process produces MVs that follow better the actual motion, so that for the same amount of bits than in the reference coding process, the encoder is



(a)

(b)

(c)

Figure 3.6: Illustrative example of the achieved subjective quality. (a) Selected part of the original frame #17 of *Mobisode*; (b) reconstructed frame with the reference software; and (c) reconstructed frame with the proposed method.

able to perform a better compression.

3.4 Conclusions

In this Chapter an intensive study on when the $\lambda_{motion}(\lambda)$ model becomes ineffective has been carried out, and an algorithm to improve the $\lambda_{motion}(\lambda)$ model and, consequently, the ME process in the RDO-based H.264/AVC video codec is proposed. Specifically, our proposal allows the encoder to choose between three different values of λ_{motion}. Actually, this choice is limited to the Inter 16x16 partition size to avoid incurring in a significant increase of the computational cost. For this partition size, the proposed algorithm allows the encoder to additionally test $\lambda_{motion} = 0$ and $\lambda_{motion} \to \infty$, which corresponds to minimum distortion and minimum rate solutions, respectively. By testing these two extreme values, the algorithm avoids to make large ME errors in ME-compromising events, which refer to a wide set of content-related events that make the block-matching model in the ME process to perform poorly; for example: complex or non translational movement, edited transitions such as *fades*, *blurring*, etc.

The proposed algorithm has been extensively tested with respect to the H.264/AVC reference software and a state-of-the-art algorithm called CALM [Zhang et al., 2010], which suggests a context adaptive adjustment of λ_{motion} to improve coding efficiency. Furthermore, the comparative assessment has been performed on a varied set of video segments exhibiting ME-compromising events to show the performance of the proposed algorithm in these cases.

The experimental results allow us to conclude that the proposed algorithm substantially improves the performance of the ME process (when *Intra modes in Inter Frames* are disabled), achieving average bit-rate reductions of -9.27% (0.52 dB in quality gain) with respect to the reference software, while the CALM algorithm achieves a bit-rate reduction of 0.13% (0.02 dB in quality gain). When considering

the overall coding efficiency, the performance improvement is lower because the *Intra modes in Inter Frames* actually compensate for some of the ME errors; nevertheless, the performance improvement is still significant: an average bit-rate reduction of -2.20% with respect to the reference software (0.12 dB in terms of quality gain); while CALM does not achieve any improvement.

Furthermore, it has been experimentally tested the effectiveness of each of the two additional λ_{motion} values, concluding that both are equally important.

Finally, two illustrative examples of the improved subjective quality achieved by the proposed algorithm have also been provided.

Chapter 4

Lagrange Multiplier Selection for Mode Decision in HEVC

In this chapter, all the contributions made to the rate-distortion optimization (RDO) problem under the HEVC standard are described.

After a preliminary analysis on both the $\lambda(QP)$ and the $\lambda_{motion}(\lambda)$ relationships, higher room for improvement in terms of coding performance was found in the revision of the $\lambda(QP)$ relationship, which was found to be inaccurate when the video content shows static backgrounds.

Then, we propose a method based on some coding-derived features concerning the sum of absolute differences (SAD) between the current and the previous frame, which adaptively decides whether a frame has static background or not and computes a proper λ value, with a minimal amount of computing time increment.

This proposal has been tested over several video sequences and compared with two versions of the HEVC reference software, HM12.0 [Bossen et al., 2013] and HM16.0 [McCann et al., 2014], and a state-of-the-art Lagrange multiplier selection algorithm [Zhao et al., 2013]. This work has been submitted for publication [González-de Suso

et al., 2016] and is currently under review.

This chapter is organized as follows: in Section 4.1, the motivation of the work is described relying on a preliminary analysis of the $\lambda(QP)$ and the $\lambda_{motion}(\lambda)$ relationships. Once the $\lambda(QP)$ relationship is pointed out as the most promising for further work, the correlation between static background and the inaccuracy of the $\lambda(QP)$ model is revealed. In Section 4.2, the proposed method is described, providing some insight into the design and parametrization of each module. In Section 4.3, we describe the experimental setup and the experiments conducted to assess the performance of the proposed method in comparison with two different versions of the HEVC reference software (HM12.0 and HM16.0) and a state-of-the-art algorithm. Finally, in Section 4.4, some conclusions are drawn.

4.1 Motivation

4.1.1 Evaluation of the Lagrangian parameter model of HEVC

In this section, the Lagrangian parameter model of HEVC is tested in a variety of situations in order to find leads for improvement. To that end, we have proposed parametrized versions of both $\lambda(QP)$ and $\lambda_{motion}(\lambda)$ relationships. Specifically, a parameter F is used to obtain different versions of the original $\lambda(QP)$ relationship; and, similarly, a parameter F_{motion} is used for $\lambda_{motion}(\lambda)$. These parametrized models are tested for a wide range of parameters over 6 CIF video sequences using the coding conditions summarized in Table 4.1.

Following the same strategy used in Chapter 3, we aim to find either an average coding performance improvement by means of a modified version of the reference relationship under study, or an improved coding performance for a subset of video

Table 4.1: Summary of coding conditions

Parameter	**Value**
GOP structure	*low-delay-P*
fps	30
QP values	[22, 27, 32, 37]
F	[0.5 : 0.4 : 2.1]
F_{motion}	[0.5 : 0.4 : 2.1]

sequences that share common visual features (i.e. motion type, texture, background, etc.).

Let us start by analyzing the $\lambda(QP)$ relationship, which is parametrized as follows from its original form in (2.16):

$$\lambda = F \cdot \alpha \cdot W_t \cdot 2^{(QP-12)/3}, \tag{4.1}$$

where F is the multiplying factor used to modify the relationship between λ and the quantization parameter (QP) factor, α is a parameter whose value depends on the frame coding type and the reference level, and W_t depends on the encoding configuration (*random-access* or *low-delay* conditions) and the hierarchy level of the frame within a group of pictures (GOP).

The coding performance achieved for each value of the F parameter in a wide range is compared with the reference encoding using the HM16.0 reference software, which corresponds to $F = 1$. To this purpose, both $\Delta R(\%)$ to measure the increment in output bit-rate and ΔY(dB) to measure the increment in objective visual quality considering the luma component are computed using the procedure described in [Bjontegaard, 2001] and [Bossen, 2013]. Obtained results are shown in Table 4.2.

It can be seen from the average performance values that increasing the F value may lead to improvements in terms of coding performance, reaching a maximum

Table 4.2: Coding performance results for several CIF video sequences and several values of F.

Video Sequence	$F = 0.5$		$F = 0.9$		$F = 1.3$		$F = 1.7$		$F = 2.1$	
	$\Delta R(\%)$	ΔY(dB)	$\Delta R(\%)$	ΔY(dB)	$\Delta R(\%)$	ΔY(dB)	$\Delta R(\%)$	ΔY(dB)	$\Delta R(\%)$	ΔY(dB)
Akiyo	12.45	-0.46	1.83	-0.07	-4.34	0.18	-7.79	0.33	-9.53	0.41
Coastguard	-0.45	0.02	-0.17	0.01	1.42	-0.04	2.83	-0.09	3.66	-0.11
Foreman	2.32	-0.09	0.26	-0.01	-0.95	0.04	-1.58	0.06	-1.85	0.08
Ice Age	0.57	-0.03	0.00	0.00	0.71	-0.04	2.56	-0.14	3.62	-0.20
Nature	2.33	-0.10	-0.38	0.02	0.81	-0.04	2.75	-0.13	5.61	-0.25
News	7.81	-0.38	1.21	-0.06	-2.58	0.13	-4.94	0.25	-6.50	0.33
Average	**4.17**	**-0.17**	**0.46**	**-0.02**	**-0.82**	**0.04**	**-1.03**	**0.05**	**-0.83**	**0.04**

of -1.03% of bit-rate savings (0.05 dB in quality gain) for $F = 1.7$. Moreover, a more detailed look allows us to notice that this improvement comes from a particular subset of video sequences such as *Akiyo*, *Foreman* and *News*. The rest of the video sequences actually show a decrement in coding performance when using a λ value different from the reference one.

After further analyzing these 3 video sequences by considering their visual features, it is clear that they all show a static background at some extent. This is specially remarkable in *Akiyo* and *News*, where a news broadcast is shown, correlating well with the higher improvement in coding performance for these sequences comparing with the rest of them. In the case of *Foreman*, although there are parts of the video sequence that show movement, there are also some parts where a static background is present. *Ice Age* and *Nature* also show a static background, but since there are *fade* transitions in the first and a *deblurring* effect which affects the entire frame in the second, it can be concluded that they cannot be characterized as having a static background in a narrow sense.

The same procedure is followed with the $\lambda_{motion}(\lambda)$ relationship in order to assess its robustness. As in the previous case, the relationship is varied with respect to the

Table 4.3: Coding performance results for several CIF video sequences and several values of F_{motion}.

Video Sequence	$F_{motion} = 0.5$		$F_{motion} = 0.9$		$F_{motion} = 1.3$		$F_{motion} = 1.7$		$F_{motion} = 2.1$	
	$\Delta R(\%)$	ΔY(dB)	$\Delta R(\%)$	ΔY(dB)	$\Delta R(\%)$	ΔY(dB)	$\Delta R(\%)$	ΔY(dB)	$\Delta R(\%)$	ΔY(dB)
Akiyo	0.37	-0.01	-0.04	0.01	0.10	0.00	0.37	-0.01	-0.03	0.00
Coastguard	0.29	-0.01	0.08	0.00	0.18	0.00	0.26	-0.01	0.48	-0.02
Foreman	0.61	-0.02	-0.23	0.01	0.40	-0.02	0.78	-0.03	1.97	-0.08
Ice Age	0.38	-0.02	0.21	-0.01	0.43	-0.02	0.56	-0.03	0.88	-0.05
Nature	0.19	-0.01	0.02	0.00	0.06	0.00	-0.22	0.01	-0.11	0.00
News	0.71	-0.04	-0.21	0.01	0.57	-0.03	0.41	-0.02	1.47	-0.07
Average	**0.42**	**-0.02**	**-0.03**	**0.00**	**0.29**	**-0.01**	**0.36**	**-0.01**	**0.78**	**-0.04**

reference version (2.14) by means of a multiplying factor:

$$\lambda_{motion} = F_{motion} \cdot \sqrt{\lambda}, \tag{4.2}$$

where F_{motion} is the multiplying factor that allows us to parametrize changes in the relationship. Note that when $F_{motion} = 1$, the reference relationship is used.

This F_{motion} factor is varied in the same manner as F before (see Table 4.1), comparing the coding performance achieved for each F_{motion} value with that obtained for the reference one. The results are shown in Table 4.3.

In this case, the average performance values show that the reference relationship is actually robust among different video sequences, showing a negligible improvement in coding performance for $F_{motion} = 0.9$, with -0.03% bit-rate savings (no gain or losses in terms of objective quality).

Thus, the robustness on the $\lambda_{motion}(\lambda)$ relationship and the room for improvement found when varying the $\lambda(QP)$ relationship for video sequences such as *Akiyo*, *Foreman* and *News* led us to perform a further analysis on the latter. Since the static background video sequences were the ones for which the improvements were observed, we suggest to further explore on previously tagged static and dynamic background video sequences to find leads for improving the $\lambda(QP)$ relationship.

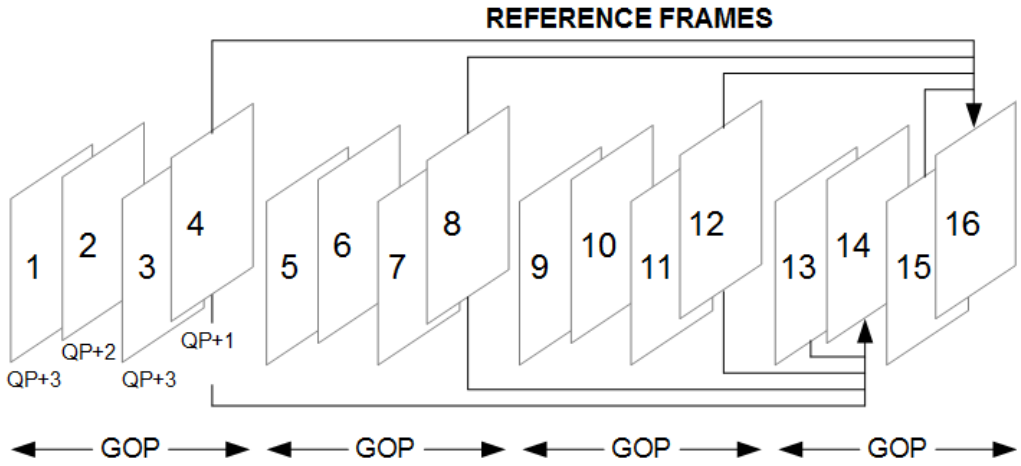


Figure 4.1: Group of pictures structure for prediction under a *low-delay-P* profile. References for frames 14 and 16 are shown.

4.1.2 A deeper analysis of the $\lambda(QP)$ relationship

In this section, to establish the motivation of our work, we first analyzed experimentally the robustness of the relationship between λ and QP proposed for HEVC over a set of video sequences with either a static or a dynamic background.

For that purpose, experiments were carried out over a set of 5 CIF and 5 HD video sequences for a *low-delay-P* profile using several values of F. A *low-delay-P* profile is suitable for static background sequences, as motion estimation is performed based on previous reference frames (as shown in Figure 4.1) [Zhang et al., 2014]. In order to draw reliable conclusions we have created *toy-examples* of short video segments of only 20 frames, so that they can be considered stationary (i.e., 20 frames of purely static or dynamic background). Moreover, a balanced number of static and dynamic background sequences has been chosen[1].

The encoder configuration used for these experiments is shown in Table 4.4. The *QP cascading* parameter refers to the frame-to-frame QP adaptation illustrated in

[1]Hereafter, the terms *static* and *dynamic* will be used referring to static background video sequences and dynamic background sequences, respectively.

Table 4.4: Encoder configuration

Parameter	**Value**
#Frames	20
QP	22, 27, 32, 37
Profile	*Low-delay-P*
QP cascading	Off
IP	-1
F Range	[0.2, 9.0]
F Step	0.4

Figure 4.1 (see frames #1 to #4); IP stands for Intra Period, which is the number of frames between Intra frames. To facilitate the study of the $\lambda(QP)$ relation, the *QP cascading* scheme was switched off, i.e., all the experiments were conducted at constant QP, and the IP was set to -1, which means that only the first frame is coded as Intra. The bit-rate increment ($\Delta R(\%)$) and objective visual quality increment (ΔY(dB)) (as defined in [Bjontegaard, 2001] and calculated following the procedure in [Bossen, 2013]) were used for assessing the coding performance in terms of bit-rate savings ($\Delta R < 0$) and quality improvement on the luma component ($\Delta Y > 0$), while the time increment $\Delta T(\%)$ was used to evaluate the computational cost:

$$\Delta T = \frac{T_{method} - T_{HM16.0}}{T_{HM16.0}} \cdot 100, \tag{4.3}$$

where T_{method} is the encoding time associated with the method under evaluation and $T_{HM16.0}$ is the encoding time of the reference encoder.

Table 4.5: Coding performance for several F values in terms of $\Delta R(\%)$ and ΔY (dBs) with respect to the reference HM16.0 software.

		$F = 0.2$		$F = 1.8$		$F = 3.4$		$F = 5.0$	
	Tag	$\Delta R(\%)$	ΔY **(dB)**	$\Delta R(\%)$	ΔY **(dB)**	$\Delta R(\%)$	ΔY **(dB)**	$\Delta R(\%)$	ΔY**(dB)**
Akiyo (CIF)	*static*	42.06	-1.56	-8.93	0.45	-13.38	0.72	-13.94	0.77
Foreman (CIF)	*static*	21.15	-0.76	-3.36	0.14	-4.40	0.19	-3.13	0.13
Ice Age (CIF)	*static*	48.83	-2.52	-1.42	0.09	-1.67	0.10	-1.54	0.09
News (CIF)	*static*	25.39	-1.26	-5.01	0.28	-7.97	0.47	-8.51	0.50
Controlled Burn (HD)	*static*	41.20	-1.44	-8.97	0.40	-14.21	0.67	-15.14	0.74
Snow Mountain (HD)	*static*	36.05	-1.25	-9.65	0.38	-15.71	0.67	-17.21	0.77
Average	***static***	**35.68**	**-1.46**	**-6.22**	**0.29**	**-9.56**	**0.47**	**-9.91**	**0.50**
Coastguard (CIF)	*dynamic*	12.18	-0.56	2.22	-0.09	4.42	-0.16	7.79	-0.24
Pedestrian (HD)	*dynamic*	11.16	-0.43	0.29	-0.01	3.49	-0.16	5.92	-0.27
Park Run (HD)	*dynamic*	12.34	-0.60	0.92	-0.05	3.31	-0.15	9.19	-0.36
Speed Bag (HD)	*dynamic*	6.02	-0.15	1.67	-0.07	6.71	-0.29	11.06	-0.47
Average	***dynamic***	**10.42**	**-0.29**	**1.27**	**-0.05**	**4.48**	**-0.19**	**8.49**	**-0.33**

4.1.2.1 Influence of the Lagrange multiplier on coding performance

Results in terms of bit-rate savings and visual quality improvement, obtained for a representative subset of the evaluated F *multipliers*, are shown in Table 4.5 (in fact, a wider range of F values were tested, but for brevity reasons only the most relevant subset is analyzed in this section). As can be observed, for some video sequences the coding performance improves with larger values of F, achieving bit-rate savings of up to -17.21% (or ΔY (dB) increments of 0.77 dBs) for *Snow Mountain*. Specifically, we observe that the coding performance improvement happens for those video sequences with static background. For instance, *Akiyo* and *News* show a news broadcast where the anchors move slightly while the background remains static.

From an optimization point of view, this improvement is due to the fact that the notable reductions in the R term for high F values exceed the corresponding small increments in the D term. Figure 4.2 illustrates the explanation of it. Let A and B be two operating points of the $R - D$ space. Given a λ value, the best coding option is that of the $R - D$ space which hits the straight line with slope λ. Consequently, when incrementing λ ($\lambda > \lambda_{ref}$) a different coding option is selected (B instead of A). In particular, since increasing λ emphasizes the weight of R in (2.5), the operating

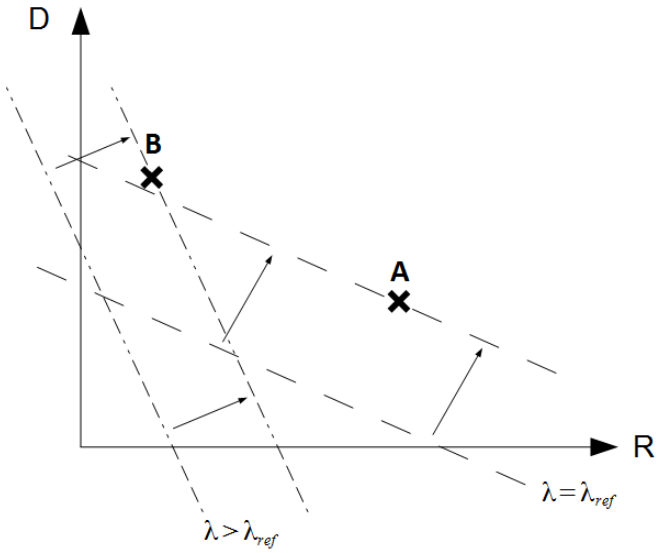


Figure 4.2: Selection of different $R - D$ points by using different λ values.

point B accounts for a lower R and a higher D.

In the particular case of *static* video sequences, these notable reductions in the R terms happen because the temporal prediction is more accurate, and coding with larger CB sizes saves lots of bits in terms of headers, indexes, etc. On the contrary, when considering *dynamic* video sequences, higher F values produce losses in video coding performance (11.06% bit-rate loss for *Speed Bag* is the worst case). Furthermore, also lower F values produce worse results than the reference value (all the F values produce positive bit-rate increments with respect to $F = 1$). As a result, we decided to code *dynamic* sequences using the baseline model. Hence, there is a need to determine in advance the type of background we are dealing with in order to either using large F values in case of static backgrounds or keep the baseline $\lambda(QP)$ relation ($F = 1$) in case of dynamic backgrounds.

Turning to *static* video sequences again, it should be noted that the optimum value of F is different from one sequence to another. Furthermore, we have considered stationary sequences (20-frame duration) in our experiments, but these conditions

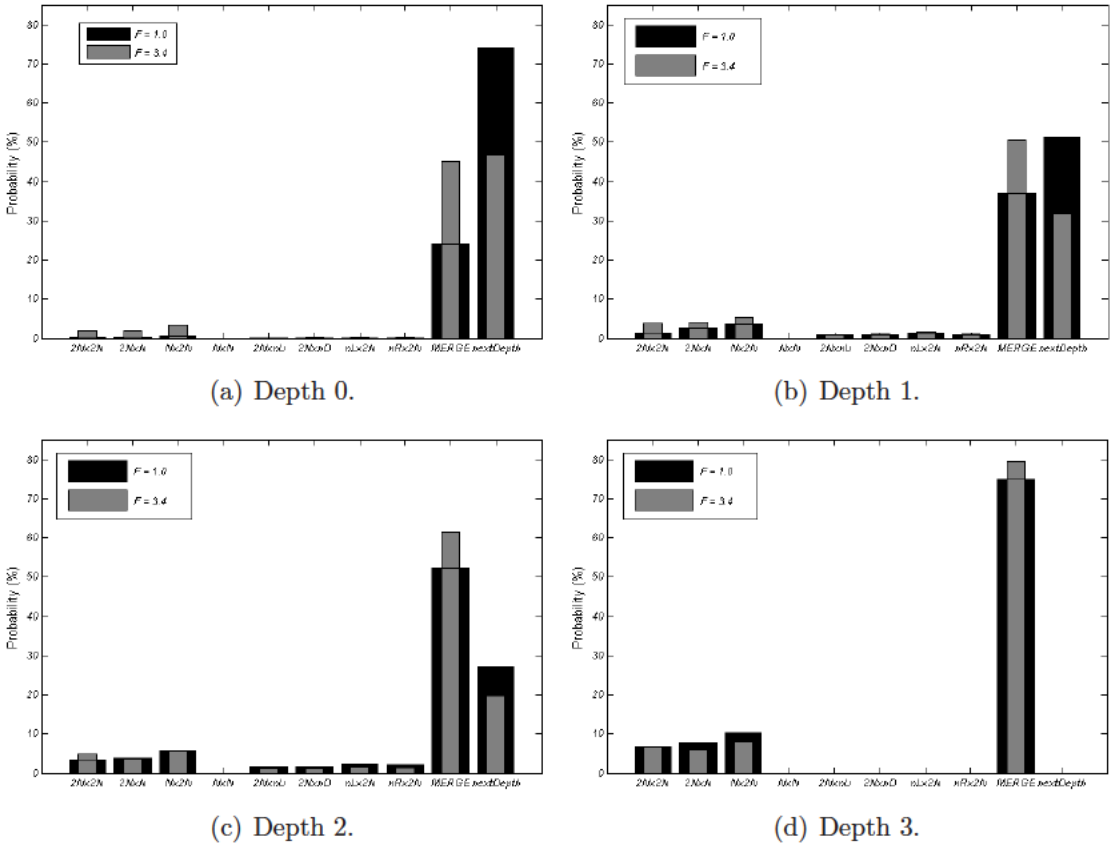
Table 4.6: Coding performance for several F values in terms of $\Delta T(\%)$ with respect to the reference HM16.0 software.

		$F = 0.2$	$F = 1.8$	$F = 3.4$	$F = 5.0$
	Tag	$\Delta T(\%)$	$\Delta T(\%)$	$\Delta T(\%)$	$\Delta T(\%)$
Akiyo (CIF)	*static*	13.03	-1.58	-3.42	-2.32
Foreman (CIF)	*static*	4.50	-1.56	-1.65	-1.60
Ice Age (CIF)	*static*	1.60	-0.61	0.11	0.10
News (CIF)	*static*	3.02	-0.26	-1.90	-2.63
Controlled Burn (HD)	*static*	17.37	-3.44	-6.06	-7.19
Snow Mountain (HD)	*static*	15.43	-3.82	-5.42	-5.35
Average	***static***	**10.99**	**-1.88**	**-3.06**	**-3.16**
Coastguard (CIF)	*dynamic*	5.40	-1.30	1.53	-3.73
Pedestrian (HD)	*dynamic*	1.70	-0.33	-0.24	0.13
Park Run (HD)	*dynamic*	5.56	1.98	4.59	5.96
Speed Bag (HD)	*dynamic*	4.27	-0.91	-2.19	-2.88
Average	***dynamic***	**4.23**	**-0.14**	**0.92**	**-0.13**

do not hold in real videos where scene changes, camera motions, or changes in the background/foreground proportion happen. Hence, an algorithm able to estimate dynamically a proper F value would be desirable.

4.1.2.2 Influence of the Lagrange multiplier on complexity

Regarding complexity, considered in Table 4.6 through the encoding time increment $\Delta T(\%)$ defined above, it can be seen that increasing F results in computational cost reductions for all static background video sequences. The reason can be found in Figure 4.3, where the probabilities of choosing an specific coding block size when encoding *Controlled Burn* (a static background sequence) at QP27 for $F = 1$ and $F = 3.4$ are shown. Specifically, each graph shows the probability of each CB size for a depth value (from 0 to 3, this value represents the depth reached by performing quad-tree divisions following the coding tree block (CTB) structure described in Section 2.1.1.2), where *nextDepth* refers to the probability of selecting a size of a deeper depth. As can be seen, the reference software ($F = 1$) selects higher depths,



(a) Depth 0.

(b) Depth 1.

(c) Depth 2.

(d) Depth 3.

Figure 4.3: Comparison between CB size probabilities for several depth values and two values of F ($F = 1$ and $F = 3.4$) for the sequence *ControlledBurn_720p30* at QP27.

while lower depths tend to be used for higher Fs. Additionally, the *merge* mode, which is a coding mode where the current CB is coded using the coding options $\overline{\theta}^*$ (as defined in Section 2.2.2) of neighboring blocks, is also more likely for higher Fs. These two facts together with the Fast Decision for Merge RD-cost of the HM16.0 reference software, which is a module that saves computations when coding with the *merge* mode [McCann et al., 2014], generate computation savings for higher Fs.

For *dynamic* video sequences, where $F = 1$ has been determined to be the optimal selection, there are no differences with respect to the baseline and the same behavior as the reference model in terms of CB size selection is expected and ΔT values are

close to zero. Additionally, note that this behavior in terms of selecting as best lower depths using $F > 1$ correlates with the $R-D$ oriented explanation done in the coding performance analysis.

4.2 Proposed Method

4.2.1 Overview

The proposed method can be described through the following steps: (i) to obtain features that describe the background of the video sequence, (ii) to classify, using these features, between *static* and *dynamic* sequences; and (iii) to find a relation between the features and the optimal F value in order to maximize gains in coding performance. Furthermore, the design of the features, the classifier, and the F estimation method should be done so that the method operates in an adaptive way and does not incur increments in computational complexity.

The flowchart of the proposed method is summarized in Figure 4.4. In the *Initialization* stage, F in (4.1) is set to 1. Then, for each frame, the corresponding features are extracted. Next, a *Classification* stage determines whether the frame is *static* or *dynamic*. If it is classified as *dynamic*, the F multiplier is set to 1; otherwise, a *Regression* stage, which also relies on the previously extracted features, is run to estimate a suitable F multiplier. Then, this F is used to encode the next frame and the process is re-run for each new frame until the end of the video sequence. Two points should be noted: (i) the proposed algorithm makes decisions on a frame basis, starting from the second encoded frame, which makes it adaptive to changes in the video sequence from the very beginning; and (ii) no training is required during the encoding process because the *Classification* and *Regression* stages are defined off-line.

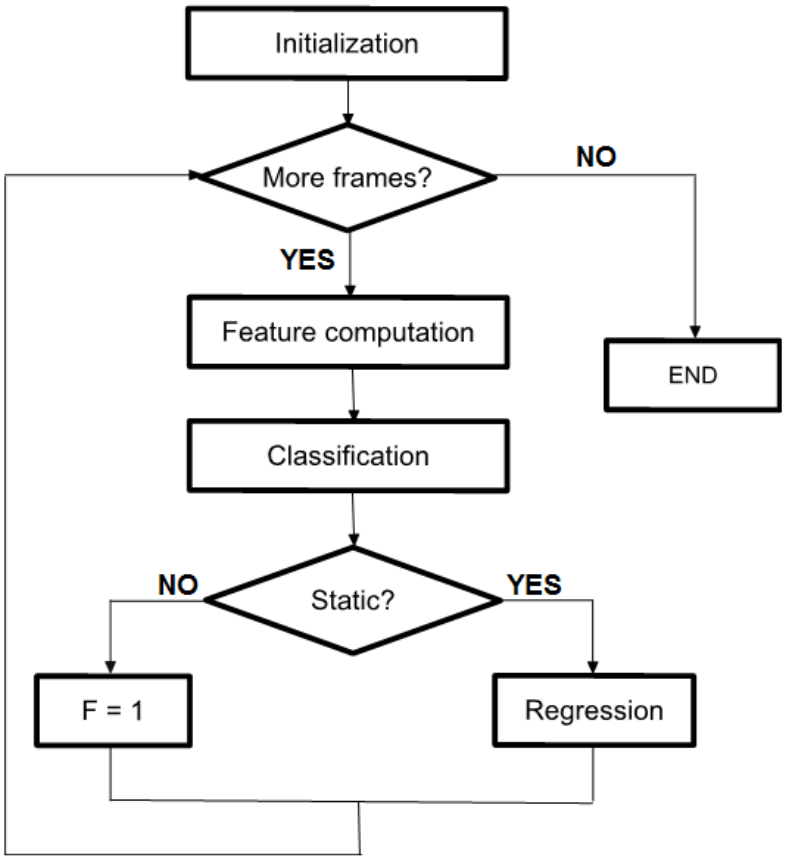
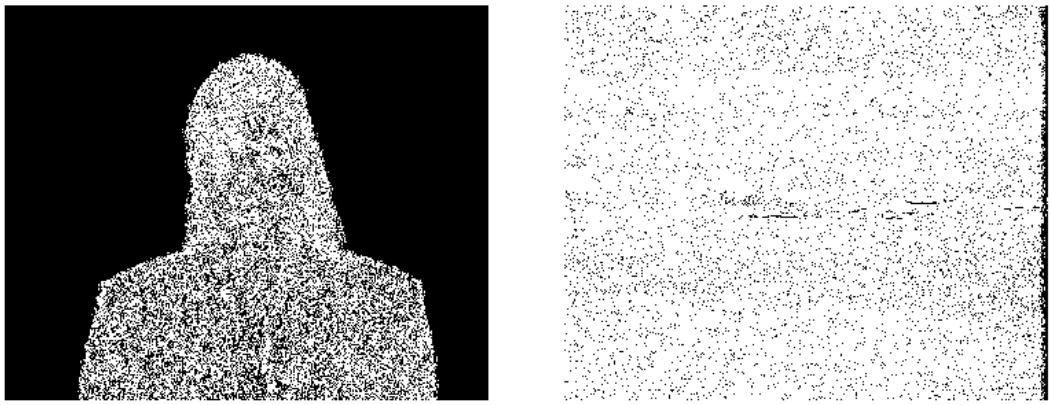


Figure 4.4: Flowchart of the proposed algorithm.

4.2.2 Feature selection

In order to find features that allow classifying each frame as either *static* or *dynamic* frame, all the video sequences used in Section 4.1 were tagged according to the motion properties of their background as *static* or *dynamic*. Then, a set of motion-related features such as the number of non-zero residual transformed coefficients, the motion vectors, or the absolute difference between pixels of different frames were tested to check if any allowed us to accurately differentiate between static or dynamic backgrounds.

Among all the analyzed features, the absolute difference between one frame and the previous one was found to be the most useful, as it is sensitive to any relative



(a) Akiyo (*static* background). (b) Coastguard (*dynamic* background).

Figure 4.5: Absolute difference images between frames #2 and #3.

movement. It should be noticed that we aim to detect static backgrounds; therefore, our feature should be sensitive to any type of movement. An illustration of this idea is shown in Figure 4.5, where binarized absolute difference images from *Akiyo* (tagged as *static*) and *Coastguard* (*dynamic*) are shown (white pixels represent high feature values and black ones represent low values). As can be seen, the static background of *Akiyo* produces nearly-zero absolute difference values, while higher values are obtained for the anchor. In *Coastguard*, almost the whole frame produces high absolute difference values, as expected from a non-static background.

To be more precise, for practical reasons, we rely on the sum of absolute differences (SAD) between the current 64x64 CB and the co-located one in the previous frame, which has been optimized to be efficiently calculated in the reference encoder and is obtained as follows:

$$SAD = \sum_{x=1}^{S_{CB}} \sum_{y=1}^{S_{CB}} |I_n(x,y) - I_{n-1}(x,y)|, \tag{4.4}$$

where $I_n(x,y)$ denotes the pixel value at the location (x,y) in the current frame, $I_{n-1}(x,y)$ denotes the same pixel in the previous frame, and S_{CB} is the maximum

CB size (which was set to 64). Another important advantage of using the SAD is that it only depends on the maximum CB size and it is independent of the encoder configuration parameters (QP, GOP structure, etc.).

As our classifier works on a frame-by-frame basis, we define the mean, SAD_m, and the standard deviation, SAD_d, of the SADs:

$$SAD_m = \frac{1}{J}\sum_{j=1}^{J} SAD_j, \tag{4.5}$$

$$SAD_d = \sqrt{\frac{1}{J-1}\sum_{j=1}^{J} (SAD_j - SAD_m)^2}, \tag{4.6}$$

where J is the number of CTUs in a frame. Additionally, SAD_m and SAD_d are normalized by their mean and standard deviation values μ_{SAD_m} and σ_{SAD_m} (resp. μ_{SAD_d} and σ_{SAD_d}) using:

$$\overline{SAD}_m = \frac{SAD_m - \mu_{SAD_m}}{\sigma_{SAD_m}}, \tag{4.7}$$

$$\overline{SAD}_d = \frac{SAD_d - \mu_{SAD_d}}{\sigma_{SAD_d}}, \tag{4.8}$$

where $\overline{SAD}_m$ and $\overline{SAD}_d$ are the normalized versions of SAD_m and SAD_d, respectively.

In order to prove that the previous features are suitable to make a correct classification, we represent in Figure 4.6 $\overline{SAD}_d$ versus $\overline{SAD}_m$ for every frame k of the considered video sequences. From data in Figure 4.6, two conclusions can be drawn. First, it is feasible to design a classifier that distinguishes between *static* and *dynamic* video sequences on this feature space. Second, both $\overline{SAD}_m$ and $\overline{SAD}_d$ features are needed to solve the problem properly since relying only on $\overline{SAD}_m$ some of the frames

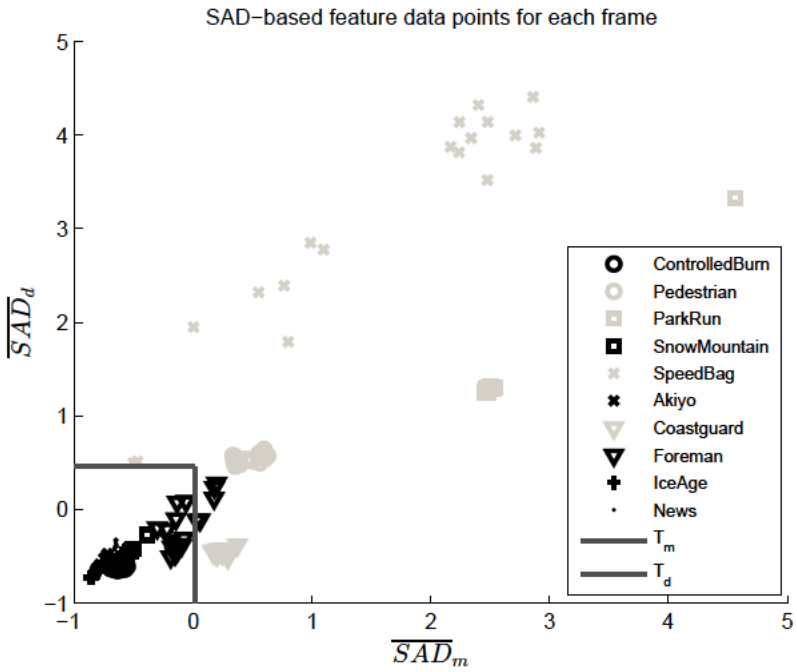


Figure 4.6: 20 frames of every video sequence are represented in the feature space defined by $\overline{SAD}_m$ and $\overline{SAD}_d$. Black points refer to *static* video sequences and gray points refer to *dynamic* video sequences. T_m and T_d represent the thresholds obtained to perform the classification.

of *Speed Bag*, for example, would be misclassified and, by relying only on $\overline{SAD}_d$, all the frames of *Coastguard*, for example, would be classified as *static*.

4.2.3 Classification

The main goal of this stage is to determine whether a frame has a *static* or a *dynamic* background. This decision turns out to be critical for the suitable operation of the proposed method since the λ value should be changed only for *static* frames to avoid losses in coding performance. The classifier, which operates on a frame basis, relies on two thresholds (on $\overline{SAD}_m$ and $\overline{SAD}_d$) to make its decision for each frame k according

to this expression:

$$C(\overline{SAD}_m^{(k)}, \overline{SAD}_d^{(k)}) = \begin{cases} static & \text{if} \quad \overline{SAD}_m^{(k)} < T_m \text{ and } \overline{SAD}_d^{(k)} < T_d \\ dynamic & \text{otherwise} \end{cases} \tag{4.9}$$

where $C(\overline{SAD}_m^{(k)}, \overline{SAD}_d^{(k)})$ represents the classifying function, T_m is the threshold on $\overline{SAD}_m$ and T_d is the threshold on $\overline{SAD}_d$.

These thresholds were obtained by evaluating the likelihood over the estimated probability distributions of both $\overline{SAD}_m$ and $\overline{SAD}_d$, given the tags *static* and *dynamic*. Specifically, for the case of the parameter $\overline{SAD}_m$, the threshold T_m was selected as the value of $\overline{SAD}_m$ which satisfies the following equation:

$$\frac{P(\overline{SAD}_m/static)}{P(\overline{SAD}_m/dynamic)} = \frac{P(dynamic)}{P(static)}, \tag{4.10}$$

where $P(\overline{SAD}_m/static)$ and $P(\overline{SAD}_m/dynamic)$ are the likelihoods of obtaining $\overline{SAD}_m$ given that the frame is either *static* or *dynamic*, respectively; and $P(static)$ and $P(dynamic)$ are the *a priori* probabilities of *static* and *dynamic*. $P(\overline{SAD}_m/static)$ and $P(\overline{SAD}_m/dynamic)$ were estimated through normalized histograms [Bishop et al., 2006]; and $P(static)$ and $P(dynamic)$ were fixed to 0.6 and 0.4, respectively, considering the number of video sequences belonging to each category. The same procedure was used to obtain T_d, finally obtaining $T_m = 0.009$ and $T_d = 0.463$. Figure 4.6 shows the classification performance in the training set when selected T_m and T_d are used.

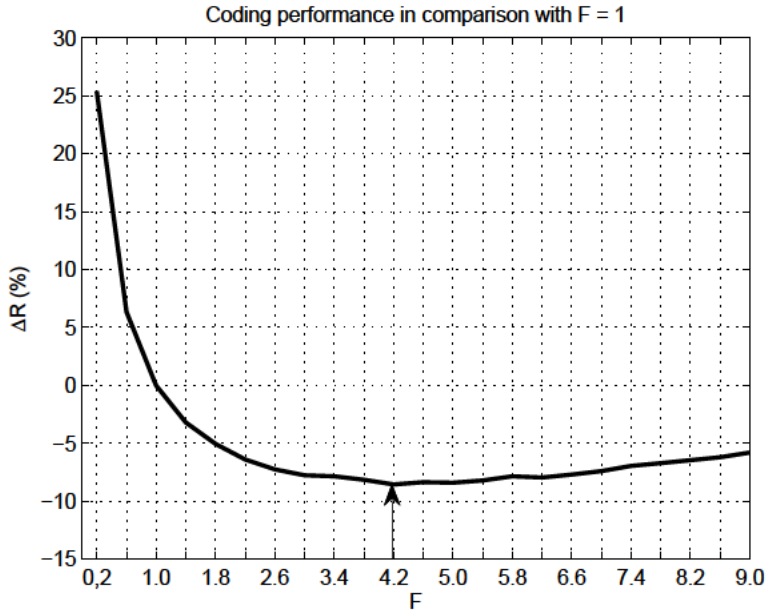


Figure 4.7: Relative coding performance (in terms of ΔR) with respect to the baseline λ ($F = 1$) as a function of F for *News* video sequence. The arrow points out the optimum value of F.

4.2.4 Regression

The main goal of this stage is to estimate a suitable value for the F multiplier in (4.1) to maximize the improvement in terms of coding efficiency.

For this purpose, all the data gathered in Section 4.1 for *static* sequences and for a wide range of F values were used as training set. First, F_{opt}, which is the value that minimized the $\Delta R(\%)$ with respect to the use of the baseline λ value ($F = 1$), was found for every sequence. An example of the achieved results is shown in Figure 4.7 for *News*. As can be seen, the $\Delta R(\%)$ reaches a minimum at $F = 4.2$, which performs notably better than the reference value. Thus, F_{opt} in this case was set as $F_{opt} = 4.2$.

Then, once F_{opt} was obtained for every *static* video sequence, a regression model that predicted this value from the previously described features was found. As shown

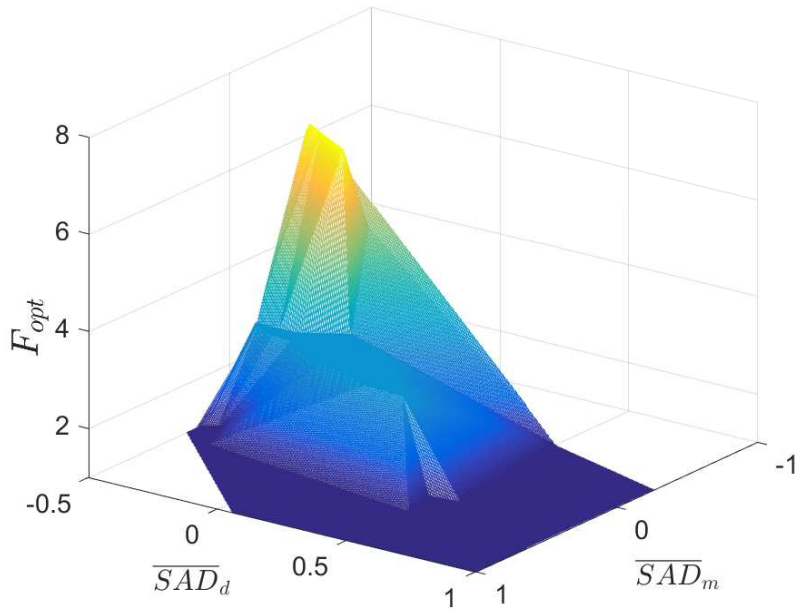


Figure 4.8: Graphical relationship between F_{opt}, $\overline{SAD}_m$ and $\overline{SAD}_d$.

in Figure 4.8, the relation between F_{opt} and $\overline{SAD}_m$ and $\overline{SAD}_d$ can be approximately modeled by means of an exponential function. Thus, the proposed frame basis estimation $\widehat{F}^{(k)}$ of F_{opt} follows the next expression:

$$\widehat{F}^{(k)} = e^{(\alpha\overline{SAD}_m^{(k)} + \beta\overline{SAD}_d^{(k)} + \delta)}, \tag{4.11}$$

where α, β and δ are regression parameters that are found by converting the previous expression into linear:

$$\ln(\widehat{F}^{(k)}) = \alpha\overline{SAD}_m^{(k)} + \beta\overline{SAD}_d^{(k)} + \delta, \tag{4.12}$$

and minimizing the mean squared error (MSE) between $\widehat{F}^{(k)}$ and F_{opt} [Bishop et al., 2006] as follows:

$$w^* = \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}s^*, \tag{4.13}$$

where $w^* = [\alpha, \beta, \delta]$ is the optimal solution; $\mathbf{X} = \left[1\ \overline{SAD}_m^{(1)}\ \overline{SAD}_d^{(1)}; 1\ \overline{SAD}_m^{(2)}\ \overline{SAD}_d^{(2)}; \cdots; 1\ \overline{SAD}_m^{(K)}\ \overline{SAD}_d^{(K)}\right]$ the extended matrix gathering training data; and $s^* = \left[\ln(F_{opt}^{(1)});\ \ln(F_{opt}^{(2)}); \cdots;\ \ln(F_{opt}^{(K)})\right]$ is the vector collecting all the $\ln(F_{opt}^{(k)})$ values corresponding to each pair $\left[\overline{SAD}_m^{(k)}, \overline{SAD}_d^{(k)}\right]$ for $k = [1, 2, \cdots, K]$, being K the number of all frames used for the parameter training process.

Finally, to adjust those estimations producing $\widehat{F}^{(k)} < 1$, which yields bad coding performance results, the final relationship was modified to:

$$\widehat{F}^{(k)} = \max\left\{1.0, e^{(\alpha\overline{SAD}_m^{(k)} + \beta\overline{SAD}_d^{(k)} + \delta)}\right\}. \tag{4.14}$$

4.2.5 Additional processing

Some additional processing over $\widehat{F}^{(k)}$ was done to avoid sudden changes of its value from frame to frame, which was experimentally checked to negatively affect the coding performance.

First, $\overline{SAD}_m^{(k)}$ and $\overline{SAD}_d^{(k)}$ features, used in the *Classification* and *Regression* stages, were computed considering N frames instead of just the current one. In particular, each feature was computed as the average value over the $N-1$ previous frames and the current one. This procedure makes the variation of $\widehat{F}^{(k)}$ over k smoother, reducing the likelihood of sudden changes. Regarding the parameter N, a trade-off should be considered: on the one hand, a high N is desirable because it implies a smooth variation of the $\widehat{F}^{(k)}$ multiplier. On the other, using a low N allows the algorithm to quickly adapt to changes in the video content.

Second, although the features from which F is estimated have been smoothed, a clipping of the frame to frame variation of $\widehat{F}^{(k)}$ has been also added. In particular, $\widehat{F}^{(k)}$ is updated as follows:

$$\widehat{F}^{(k)} = \begin{cases} \max\left\{\widehat{F}^{(k)}, \widehat{F}^{(k-1)} - \mathrm{Th}\right\} & \text{if } \widehat{F}^{(k)} \leq \widehat{F}^{(k-1)} \\ \min\left\{\widehat{F}^{(k)}, \widehat{F}^{(k-1)} + \mathrm{Th}\right\} & \text{if } \widehat{F}^{(k)} > \widehat{F}^{(k-1)}, \end{cases} \tag{4.15}$$

where $\widehat{F}^{(k-1)}$ denotes the estimation for the previous frame and Th is the clipping threshold, which enables a better control of $\widehat{F}^{(k)}$ on a frame basis.

Proper values of N and Th ($N = 10$ and $Th = 1.5$) were selected using a set of *Train* sequences (the ones in Table 4.7), achieving -1.08% bit-rate reduction (0.04 dBs of ΔY) evaluating on the *Test* video sequences in Table 4.7 with respect to a version without this additional processing.

4.2.6 Algorithm

The complete algorithm is summarized in Algorithm 3, where the specific values for the parameters T_m, T_d, Th, N, α, β and δ are given.

Algorithm 3 Proposed coding process.

Require: K number of frames.
Require: J number of CTUs.
Require: $N = 10$.
Require: Normalizing parameters μ_{SAD_m}, σ_{SAD_m}, μ_{SAD_d}, σ_{SAD_d}.
Require: $T_m = 0.009$, $T_d = 0.463$.
Require: $\alpha = 0.62$, $\beta = 0.01$, $\delta = 1.01$.
Require: Th = 1.5.
Require: $\widehat{F}^{(0)} = 1$.

for $\forall\ k \in K$ **do**
  for $\forall\ j \in J$ **do**
    Compute SAD_j.
    Use $\widehat{F}^{(k-1)}$ in (4.1) to perform coding.
  end for
  Compute $SAD_m^{(k)}$ and $SAD_d^{(k)}$.
  Compute $\overline{SAD}_m^{(k)}$ and $\overline{SAD}_d^{(k)}$ using (4.7) and (4.8).
  if $(\frac{1}{N}\overline{SAD}_m^{(k)} + \frac{1}{N}\sum_{n=1}^{N-1}\overline{SAD}_m^{(n)} < T_m)$
    and $(\frac{1}{N}\overline{SAD}_d^{(k)} + \frac{1}{N}\sum_{n=1}^{N-1}\overline{SAD}_d^{(n)} < T_d)$ **then**
    Compute $\widehat{F}^{(k)}$ by using (4.14) and (4.15).
  else
    $\widehat{F}^{(k)} = 1$.
  end if
end for

4.3 Experimentation

In this section, we first assess the two main subsystems of the proposed method, i.e., the classifier and the regressor. Then, we evaluate the coding performance of our proposal in comparison with the HEVC standard and a state of the art method [Zhao et al., 2013]. Then, we test the capability of the proposed method to adapt to varying

video content. Finally, we provide an illustration of the subjective quality.

4.3.1 Classifier and regressor assessment

Before assessing the coding performance of the proposed method, we have checked the efficacy of its two main subsystems separately. To this purpose, we have used as *Train* set the same set of sequences used in Section 4.1 and Subsection 4.2.4 and we have added a *Test* set composed of 12 different sequences (see Table 4.7 for a complete list of sequences). The type of background for all these sequences was manually labeled and the same procedure of Subsection 4.2.4 was carried out to obtain F_{opt}.

To assess the classifier, the following accuracy measure was used:

$$A(\%) = 100 - \frac{100}{K} \cdot \sum_{n=1}^{K} |T_C - T_{GT}|, \tag{4.16}$$

where T_C is the tag provided by the classifier (being 1 for *static* videos, and 0 for *dynamic* videos), T_{GT} is the ground-truth tag listed in the third column of Table 4.7, and K the number of coded frames, which was set to 20.

The obtained results are shown in Table 4.7. An average accuracy of 93.33% was obtained on the *Test* set, being almost perfect in 11 of the 12 video sequences.

To properly assess the regressor, the proposed method was compared with an "optimal" encoder using F_{opt} (fourth column of Table 4.7) for each *static* video sequence (e.g., *Akiyo* was encoded with $F = 5.4$ and *Foreman* with $F = 2.6$).

Results in terms of visual quality and bit-rate increments, ΔY (dB) and ΔR (%), respectively, are shown in Table 4.8 with respect to the "optimal" encoder. As can be seen, the proposed method incurs a bit-rate loss of 0.75% (or a visual quality loss of -0.02 dB) for the *Test* set when compared to the "optimal" encoder. Since this performance is quite close to that of the "optimal" encoder, we consider that the

Table 4.7: Classification accuracy A(%) of the proposed method for both train and test video sequences.

Sample type	Sequence	Tag	F_{opt}	A(%)
Train	*Akiyo (CIF)*	*static*	5.4	100
	Foreman (CIF)	*static*	2.6	25
	Ice Age (CIF)	*static*	3.4	100
	News (CIF)	*static*	4.2	100
	Controlled Burn (HD)	*static*	5.8	100
	Snow Mountain (HD)	*static*	6.2	100
	Coastguard (CIF)	*dynamic*	1.0	100
	Pedestrian (HD)	*dynamic*	1.0	100
	Park Run (HD)	*dynamic*	1.0	100
	Speed Bag (HD)	*dynamic*	1.0	100
	Average	**-**	**-**	**92.50**
Test	*Bridge Close (CIF)*	*static*	4.6	100
	Bridge Far (CIF)	*static*	2.2	100
	Container (CIF)	*static*	3.8	100
	Hall (CIF)	*static*	5.4	100
	Highway (CIF)	*static*	2.6	100
	Sequence 3 (SD)	*static*	2.6	25
	Tiger & Dragon (SD)	*static*	3.4	100
	Last Samurai (SD)	*static*	3.0	100
	In To Tree (HD)	*static*	3.4	95
	Sequence 10 (SD)	*dynamic*	1.0	100
	Soccer (CIF)	*dynamic*	1.0	100
	Riverbed (HD)	*dynamic*	1.0	100
	Average	**-**	**-**	**93.33**

regressor is performing well.

Finally, it is worth noticing that although there is little room for improving the regressor, the classifier seems to be more critical: in the sequence exhibiting the highest losses in terms of $\Delta R(\%)$ (*Foreman*), the classification result is quite poor (see Table 4.7).

In summary, these results prove that: (i) the proposed classifier allows us to suitably detect static backgrounds for which to modify the $\widehat{F}^{(k)}$ multiplier; and (ii) the proposed regressor can obtain a proper $\widehat{F}^{(k)}$ estimation.

Table 4.8: Coding performance of the proposed method relative to that of an "optimal" encoder using F_{opt}.

			Proposed	
Sample type	**Sequence**	F_{opt}	ΔR (%)	ΔY **(dB)**
Train	*Akiyo (CIF)*	5.4	0.35	-0.02
	Foreman (CIF)	2.6	3.66	-0.15
	Ice Age (CIF)	3.4	0.07	0.00
	News (CIF)	4.2	0.44	-0.02
	Controlled Burn (HD)	5.8	0.70	-0.04
	Snow Mountain (HD)	6.2	0.93	-0.05
	Average	**-**	**1.02**	**-0.05**
Test	*Bridge Close (CIF)*	4.6	2.02	-0.06
	Bridge Far (CIF)	2.2	1.26	-0.02
	Container (CIF)	3.8	0.00	0.00
	Hall (CIF)	5.4	1.95	-0.09
	Highway (CIF)	2.6	1.22	-0.03
	Sequence 3 (SD)	2.6	0.36	-0.01
	Tiger & Dragon (SD)	3.4	0.05	0.00
	Last Samurai (SD)	3.0	-0.15	0.01
	In To Tree (HD)	3.4	0.07	0.01
	Average	**-**	**0.75**	**-0.02**

4.3.2 Coding performance evaluation

The proposed method was implemented in the versions HM12.0 [Bossen et al., 2013] and HM16.0 [McCann et al., 2014] of the reference software and the encoder configuration was the one shown in Table 4.9. For the coding performance evaluation, the set of video sequences has been extended with the E Sequences from the HEVC evaluation corpus in [Bossen, 2013] (*Four People*, *Kristen and Sara* and *Johnny*). Furthermore, 100 frames of every video sequence were encoded (instead of the 20 frames of previous analyses). It should be noticed that, as long as more frames have been included for these experiments, it would have been more precise to rename the types of sequences as *mainly static* or *mainly dynamic* to account for potential variations over the 100 frames. However, we have preferred to keep the original *static* vs. *dynamic* division to study the behavior of the algorithm separately when applicable.

Table 4.9: Encoder configuration for the HM12.0 and HM16.0 experiments.

Parameter	**Value**
#Frames	100
QP	22, 27, 32, 37
Profile	*Low-delay-P*
QP cascading	On
IP (in HM12.0)	-1
IP (in HM16.0)	32

4.3.2.1 Comparison with State of the Art

A first set of experiments was devoted to perform a comparison between the proposed method and a state-of-the-art method [Zhao et al., 2013], which computes a λ value for each CTU based on the proportion of static background. To that end, we used HM12.0 because the authors of [Zhao et al., 2013] kindly provided us with an executable file of their method implemented in HM12.0 and an encoding configuration file with their coding conditions, which we used. The obtained results are shown in Table 4.10.

For *static* video sequences, the proposed method achieves an average gain of -13.80% in terms of bit-rate savings (or 0.46 dBs in terms of ΔY) with respect to the reference software; while the method described in [Zhao et al., 2013] achieves an average gain of -2.46% in ΔR (0.03 dBs in ΔY). Moreover, for *dynamic* video sequences, our proposal applied the reference λ, limiting losses to 1.15% in terms of bit-rate, while the method described in [Zhao et al., 2013] (not prepared to deal with *dynamic* video sequences) incurred a bit-rate loss of 5.86%.

This notable performance difference can be explained by two main reasons: (i) the proposal in [Zhao et al., 2013] trained the model "on the fly" at the beginning of the encoding process, using M frames of the original video sequence. Thus, during these

Table 4.10: Coding performance of the proposed algorithm and [Zhao et al., 2013] relative to the HM12.0 reference software.

		Reference [Zhao et al., 2013]			**Proposed Method**		
Tag		ΔT (%)	ΔR(%)	ΔY(dB)	ΔT (%)	ΔR(%)	ΔY(dB)
static	*Akiyo (CIF)*	-3.73	-3.40	0.14	-11.88	-14.32	0.64
	Bridge Close (CIF)	-22.83	-1.15	0.03	-33.36	-22.50	0.59
	Bridge Far (CIF)	-22.38	-17.93	-0.12	-28.96	-22.55	0.08
	Container (CIF)	-10.35	-4.05	0.13	-17.54	-11.03	0.37
	Hall (CIF)	-22.40	-5.42	0.17	-28.62	-19.36	0.66
	Highway (CIF)	-19.81	1.58	-0.04	-22.23	-2.98	0.07
	Ice Age (CIF)	0.44	2.81	-0.15	-7.25	-13.89	0.83
	News (CIF)	-0.78	1.32	-0.07	-13.46	-9.45	0.47
	Last Samurai (SD)	-3.83	0.84	-0.03	-4.91	-25.88	1.01
	Tiger & Dragon (SD)	-5.19	2.00	-0.07	-8.15	1.97	-0.07
	Controlled Burn (HD)	-11.23	-5.13	0.17	-19.98	-21.45	0.83
	Four People (HD)	-3.80	0.89	-0.04	-7.57	-11.54	0.47
	In To Tree (HD)	-16.40	2.23	-0.05	-22.04	-4.60	0.08
	Kristen and Sara (HD)	-6.58	0.00	0.00	-9.51	-8.60	0.31
	Johnny (HD)	-7.28	1.13	-0.02	-8.98	-7.96	0.23
	Snow Mountain (HD)	-15.68	-15.03	0.40	-23.57	-26.64	0.87
	Average (static)	**-10.74**	**-2.46**	**0.03**	**-16.75**	**-13.80**	**0.46**
dynamic	*Foreman (CIF)*	-6.38	4.27	-0.17	-2.45	0.91	-0.04
	Coastguard (CIF)	-6.26	7.66	-0.27	-2.88	0.00	0.00
	Soccer (CIF)	-5.87	6.70	-0.28	-3.22	0.00	0.00
	Sequence 3 (SD)	-7.19	6.52	-0.24	0.20	0.12	0.00
	Sequence 10 (SD)	-3.37	5.14	-0.18	2.26	0.00	0.00
	Park Run (HD)	-7.20	4.77	-0.20	-0.88	0.00	0.00
	Pedestrian (HD)	-4.46	7.41	-0.31	-4.11	0.00	0.00
	Riverbed (HD)	-4.01	4.62	-0.23	-3.51	0.00	0.00
	Speed Bag (HD)	-6.39	5.64	-0.21	-6.29	9.43	-0.33
	Average (dynamic)	**-5.68**	**5.86**	**-0.24**	**-2.32**	**1.15**	**-0.04**
	Average (all)	**-8.92**	**0.53**	**-0.07**	**-11.55**	**-8.42**	**0.28**

M frames, [Zhao et al., 2013] did not change the λ value and thus did not achieve improvements comparing with the reference software. This approach works well for video surveillance sequences, but it does not work well for typical video sequences, where the video content changes and thus, the model becomes inefficient (because it was trained for other type of video content). To solve this problem, the model should be re-trained after such changes in the video content, using other M frames in which the algorithm is not enabled. (ii) [Zhao et al., 2013] uses a uni-dimensional space of

background percentage bins. Therefore, in some video sequences, one or more of the bins may not have enough data for the training process (as not enough CTUs with a certain percentage of background may be available). Thus, the parametrization of the relationship between background percentage and λ can be inaccurate, leading to poor results in terms of coding performance efficiency. To solve this, the algorithm would need a larger number of frames for training.

We perform both *Classification* and *Regression* parametrizations "off-line" using a bi-dimensional feature space of normalized SAD mean and standard deviations which work properly for any video content, as we have shown in Subsection 4.2.2. Therefore, our approach solves the problems previously described for [Zhao et al., 2013], significantly outperforming its performance.

In terms of the computational efficiency, the proposed method, due to reasons explained in Section 4.1.2.2, provides on average a time saving of -11.55% compared with the reference software, while the method presented in [Zhao et al., 2013] generates a time saving of -8.92%. However, it should be noted that the computational time required for the training process in [Zhao et al., 2013], which is very complex, is not taken into account in the results. Also, note that the proposed method is fully compatible with many complexity reduction and complexity control approaches in the state of the art (e.g., [Shen et al., 2013, Xiong et al., 2014, Jiménez-Moreno et al., 2016]).

Finally, considering the reference model, it is worth noticing that although the *QP cascading*, which also acts on QP on a frame basis, causes a similar effect to that of increasing the λ multiplier, the improvement in coding performance obtained by our proposal is still significantly larger. This improvement comes from the fact that *QP cascading* does not take the video content into account; while the proposed method produces a content-aware λ adaptation and, furthermore, it adapts λ in a

wider dynamic range than that of the *QP cascading*.

4.3.2.2 Comparison with HM16.0 reference software

A second set of experiments was performed to compare the proposed method with a more recent version of the reference software, namely HM16.0, and using a more common encoder configuration (the IP parameter was set to 32 for 30 frames-per-second video sequences, as recommended in [Bossen, 2013]) for general purpose video coding.

The improvements of the proposed algorithm over HM16.0 reference software are still quite significant. In particular, an average bit-rate saving of -11.07% (0.42 dBs in terms of ΔY) was achieved for *static* sequences and quite similar results than those of the reference (a bit-rate increment of 1.00%) were achieved for *dynamic* sequences. Moreover, taking into account all the sequences, an average bit-rate reduction of -6.72% (or an increment in visual quality of 0.25 dBs) was achieved. The improvements are a little bit lower when compared with those achieved with respect to HM12.0 simply because we have changed the encoder configuration to include an Intra frame every 32 frames, and Intra frames do not benefit as much as Inter frames from adapting the λ parameter.

4.3.3 Adaptive performance

In this subsection, the adaptation capability of the proposed method is assessed. To that purpose, a simple variation of the proposed method was implemented in the HM16.0 reference software. In particular, the *Classification* and *Regression* stages were only activated in the first frame, keeping the obtained $\widehat{F}^{(1)}$ multiplier constant for the rest of the video sequence. The results obtained by this variation of the

Table 4.11: Coding performance of the proposed algorithm relative to the HM16.0 reference software.

		Proposed Method		
Tag		ΔT (%)	ΔR(%)	ΔY(dB)
static	*Akiyo (CIF)*	-7.59	-10.49	0.51
	Bridge Close (CIF)	-17.57	-18.06	0.50
	Bridge Far (CIF)	-13.43	-25.25	0.32
	Container (CIF)	-9.82	-7.23	0.28
	Hall (CIF)	-17.02	-13.90	0.53
	Highway (CIF)	-15.18	0.60	-0.01
	Ice Age (CIF)	-4.75	-22.44	1.38
	News (CIF)	-8.12	-5.27	0.27
	Last Samurai (SD)	-3.19	-22.50	0.87
	Tiger & Dragon (SD)	-6.17	1.75	-0.07
	Controlled Burn (HD)	-13.08	-13.52	0.55
	Four People (HD)	-6.22	-10.13	0.44
	In To Tree (HD)	-11.91	-3.27	0.07
	Kristen and Sara (HD)	-8.30	-7.95	0.30
	Johnny (HD)	-6.81	-5.75	0.17
	Snow Mountain (HD)	-13.57	-13.75	0.54
	Average (static)	**-10.17**	**-11.07**	**0.42**
dynamic	*Foreman (CIF)*	-2.40	2.04	-0.08
	Coastguard (CIF)	0.55	0.00	0.00
	Soccer (CIF)	-1.88	0.00	0.00
	Sequence3 (SD)	0.25	0.14	-0.01
	Sequence10 (SD)	0.37	0.00	0.00
	Riverbed (HD)	0.10	0.00	0.00
	Pedestrian (HD)	0.10	0.00	0.00
	Park Run (HD)	0.48	0.00	0.00
	Speed Bag (HD)	-3.83	6.84	-0.24
	Average (dynamic)	**-0.70**	**1.00**	**-0.04**
	Average (all)	**-6.76**	**-6.72**	**0.25**

proposed method (hereafter referred to as fixed-F) are compared with those of the complete proposal in Table 4.12.

For *static* video sequences, an improvement of -4.14% in terms of bit-rate savings (0.17 dBs in terms of ΔY) was achieved by adapting the λ parameter to the video content. For *dynamic* video sequences, the method incurred reduced losses of 1.00% (-0.04 dBs) due to some misclassifications. Taking into account the whole set of sequences, a global improvement of -2.29% (0.09 dBs) was obtained.

Table 4.12: Coding performance comparison of the proposed algorithm and the fixed-F version relative to the HM16.0 reference software.

Tag		**Fixed-F Method**		**Proposed Method**	
		ΔR(%)	ΔY(dB)	ΔR(%)	ΔY(dB)
static	*Akiyo (CIF)*	-6.94	0.33	-10.49	0.51
	Bridge Close (CIF)	-10.82	0.30	-18.06	0.50
	Bridge Far (CIF)	-19.86	0.19	-25.25	0.32
	Container (CIF)	-5.41	0.22	-7.23	0.28
	Hall (CIF)	-10.43	0.39	-13.90	0.53
	Highway (CIF)	-0.69	0.02	0.60	-0.01
	Ice Age (CIF)	-13.19	0.77	-22.44	1.38
	News (CIF)	-4.01	0.21	-8.12	0.27
	Last Samurai (SD)	-5.28	0.19	-22.50	0.87
	Tiger & Dragon (SD)	-0.57	0.02	1.75	-0.07
	Controlled Burn (HD)	-8.13	0.35	-13.52	0.55
	Four People (HD)	-6.79	0.28	-10.13	0.44
	In To Tree (HD)	0.00	0.00	-3.27	0.07
	Kristen and Sara (HD)	-5.76	0.21	-7.95	0.30
	Johnny (HD)	-3.95	0.11	-5.75	0.17
	Snow Mountain (HD)	-9.13	0.37	-13.75	0.54
	Average (static)	**-6.93**	**0.25**	**-11.07**	**0.42**
dynamic	*Foreman (CIF)*	0.00	0.00	2.04	-0.08
	Coastguard (CIF)	0.00	0.00	0.00	0.00
	Soccer (CIF)	0.00	0.00	0.00	0.00
	Sequence 3 (SD)	0.00	0.00	0.14	-0.01
	Sequence 10 (SD)	0.00	0.00	0.00	0.00
	Park Run (HD)	0.00	0.00	0.00	0.00
	Pedestrian (HD)	0.00	0.00	0.00	0.00
	Riverbed (HD)	0.00	0.00	0.00	0.00
	Speed Bag (HD)	0.00	0.00	6.84	-0.24
	Average (dynamic)	**0.00**	**0.00**	**1.00**	**-0.04**
	Average (all)	**-4.43**	**0.16**	**-6.72**	**0.25**

Some of the more appealing results happen for the *Last Samurai* sequence, in which the background changes over time. In particular, along the first 100 frames three different scenarios are shown, separated by scene cuts, each exhibiting a different amount of static background. In this case, the fixed-F method achieves a ΔR improvement of -5.28% because the first scene exhibits a static background. Nevertheless, by allowing the algorithm to adapt to the content, the proposed algorithm reaches a bit-rate saving of -22.50%, which is significantly higher than that

achieved by the fixed-F method. The same behavior can be observed for *Ice Age*, where a *cross-fade* between two scenes happens at frame #85. There, the proposed method is able to properly adapt the λ parameter yielding a significant improvement in coding performance (-22.44% bit-rate saving vs. -13.19% of the fixed-F method). Furthermore, it is worth mentioning the performance improvement for *In To Tree*, which shows a movement towards a tree in a static scene. In this case, the proposed method is able to adapt to those fragments of the video sequence in which the movement is not important, achieving -3.27% coding improvements relative to both the reference HM16.0 and the fixed-F method.

Finally, it is also worth discussing the case of *Speed Bag*, where the fixed-F method achieves a notably better result than that of the proposed method. This happens because one important segment of this sequence shows illumination changes that are not properly managed by the classifier, yielding significant coding losses.

In summary, it can be concluded that the adaptation capability of the proposed method makes it to manage properly realistic situations in which background characteristics change over time.

(a)

(b)

(c)

Figure 4.9: *Controlled Burn* decoded video fragments belonging to frame #8. (up) Original sequence. (bottom left) Decoded frame using the reference HM16.0 software. (bottom right) Decoded frame using the proposed method.

4.3.4 Subjective quality assessment

In addition to the objective evaluation relying on $\Delta R(\%)$ and $\Delta Y(\text{dB})$, we provide an illustration of the subjective quality achieved. Specifically, two fragments of one frame of *Controlled Burn* and *Snow Mountain* were evaluated.

In order to properly evaluate them, the HM16.0 reference software was used to encode 20 frames belonging to both sequences at QP32, obtaining a target bit-rate for the proposed method. Then, the QP was adjusted for the proposed method to

(a)

(b)

(c)

Figure 4.10: *Snow Mountain* decoded video fragments belonging to frame #16. (up) Original sequence. (bottom left) Decoded frame using the reference HM16.0 software. (bottom right) Decoded frame using the proposed method.

produce a similar bit-rate and a subjective visual analysis was performed.

In Figure 4.9, the original frame #8 is shown on top, and this same frame is shown at the bottom when decoded having previously used the HM16.0 reference software in Figure 4.9(b) (coded at 35.7 Kbits) and the proposed method in Figure 4.9(c) (coded at 35.2 Kbits). As can be seen, the frame obtained by the proposed method shows more detail in the image than that of the reference software. It is specially noticeable in the right wall of the hut, where the horizontal lines are blurred to the point of almost disappear in the left image.

In Figure 4.10, a similar behavior is noticed in *Snow Mountain* video sequence.

In this case, frame #16 is shown, which has been coded at 37.1 Kbits in the reference software and at 37.2 Kbits in the proposed method. As can be observed, the same differences in terms of detail are noticeable in the examples. Specifically, it is worth noticing detail differences in the trees shown at the center part of the fragment.

Thus, as shown by the previous objective evaluations and illustrated through two subjective examples, the proposed method saves bits by adapting the $\lambda(QP)$ relationship, allowing the encoder to obtain a better visual quality for the same target bit-rate when compared with the reference HM16.0 software.

4.4 Conclusions

In this chapter a method has been proposed to adaptively select the Lagrangian parameter λ of the cost function associated with the RDO process in the HEVC reference software. This approach has been motivated by means of an experiment that proves that video sequences with static background are more efficiently encoded using higher values of the parameter λ than that of the reference software.

In order to determine whether the background of a sequence is static on a frame basis, some coding-derived features that describe the static or dynamic nature of the background have been found and a classifier has been designed. Furthermore, an exponential regression function has also been proposed to estimate a proper value of the λ parameter. In so doing, the proposed method becomes content-aware, being able to dynamically increase the λ parameter when encoding static background video sequences and keeping it as in the reference software when encoding dynamic background sequences.

The efficacy of both the classifier and the regressor has been experimentally proved. Subsequently, the proposed method has been compared with a state-of-

the-art method [Zhao et al., 2013] yielding a significantly better average performance. Moreover, the proposed method has also been assessed in comparison with the HM16.0 version of the reference software, achieving average bit-rate savings of -11.07% (or ΔY gains of 0.42 dBs) for *static* video sequences and incurring quite limited losses for *dynamic* sequences. All these conclusions have been supported by a comprehensive set of experiments over a large set of video sequences. Furthermore, an illustrative example of subjective improvement has been provided.

Finally, the computational efficiency of the proposed method has also been assessed, proving that in average the proposed method turns out to be less demanding than the reference software.

Chapter 5

Conclusions and further work

5.1 Conclusions

In this PhD. thesis, some contributions have been made to the rate-distortion optimization (RDO) problem in video coding. In particular, we have focused on the two most recent video coding standards, i.e., H.264/AVC and HEVC, with the aim of improving the reference versions of the Lagrangian-based RDO processes implemented in both reference encoder softwares by means of generalized models. In both cases, taking the JM15.1 (H.264/AVC) and the HM16.0 (HEVC) versions of the reference software as references, we have studied first the potential sources of inaccuracies of the respective rate-distortion models and, relying on our findings, we have proposed specific algorithms that have proved to improve the original models.

Thus, in the case of H.264/AVC, a preliminary study of its rate-distortion optimization model led us to carry out a deeper analysis of the $\lambda_{motion}(\lambda)$ relationship, concluding that unbalanced errors in the estimation of distortion and rate made in the motion estimation module produce non-optimal decisions, affecting the pair (motion vector, reference frame). This conclusion provides a more accurate interpretation of the limitations of the $\lambda_{motion}(\lambda)$ relationship than previous explanations in

the literature. Furthermore, these unbalanced estimation errors have been studied to happen more often when video content compromises the block-matching model of the encoder.

As a result of these findings, a method has been proposed that, on a macroblock basis, evaluates 2 additional modes, the minimal rate decision (representing an arbitrarily large λ_{motion}) and the minimal distortion decision ($\lambda_{motion} = 0$) which minimize either R_{motion} or D_{motion} respectively in the J_{motion} cost function evaluation, achieving significant improvements in terms of coding efficiency by selectively using them when necessary. This method has been extensively tested in different encoding conditions and both the minimal rate decision and the minimal distortion decision proposals have been validated separately, obtaining an average improvement over an state-of-the-art method of -9.27% in terms of ΔR (0.52 dB in ΔY) when considering only the motion estimation process, and -2.20% bit-rate savings when activating the Intra modes (0.12 dB in objective quality increment). Additionally, subjective quality improvements have also been shown.

In the case of HEVC, the same preliminary study was performed leading us to carry out a further study on the $\lambda(QP)$ relationship, which pointed out the static background video sequences as the main source of inaccuracy of the rate-distortion optimization model implemented in the HEVC reference software.

Taking these results into consideration, some coding-derived features were proposed in order to describe the background of the video sequence and a classifier for tagging video frames according to their static or dynamic background was designed. Moreover, a reasonable estimation of the Lagrangian parameter λ was proposed based on an exponential regressor, which has been proved to provide a notable improvement of the encoder performance on those videos exhibiting a static background.

Thus, a method that includes the classifier, the regressor and some additional

post-processing has been proposed and tested over a large set of video sequences, showing that the proposed method improves the coding performance of the reference implementation of the HEVC standard and that of a state-of-the-art method [Zhao et al., 2013] that suggests a method for adapting λ in a coding tree unit basis.

Specifically, the proposed method is more general than that of [Zhao et al., 2013] and, besides producing a -13.80% bit-rate improvements (0.46 dBs of visual quality improvements) over static background sequences, it does not incur significant coding losses when processing dynamic sequences by virtue of the proposed background classifier. Moreover, the proposed method have been tested with respect to the HM16.0 version of the HEVC reference software producing average bit-rate savings of -11.07% (0.42 dBs of visual quality increment) over static background sequences. Furthermore, the adaptive performance of the proposed method has been validated with respect to a non-adaptive version of the proposed method. Finally, besides providing a conclusive set of objective results, we have shown a couple of examples where the improvement is subjectively evident.

It is important to highlight that the proposed methods are standard-compliant, as they only affect the rate-distortion optimization process, and they are easy to implement in the reference software, so they are susceptible to being incorporated on future encoder implementations.

5.2 Further work

As future lines of research, the study on the Lagrangian models proposed in this thesis has been limited to P-frame-based temporal structures; thus, additional studies considering B-frames would be interesting. In the H.264/AVC standard, on the one hand, IPxB structures that perform bi-prediction independently on the past and

future reference frames are expected to behave in a similar manner than IPPP structures. On the other hand, sequential prediction using both past and future references jointly may arise new limitations of the reference model. For HEVC, the *low-delay-B* configuration is also expected to behave as *low-delay-P*, but *random-access* configuration will surely behave different, as future references are used. Thus, further analysis of the Lagrangian models on these temporal structures might also reveal new limitations of the model.

Next, we propose specific future lines of research for each proposed method. In H.264/AVC, a promising approach would be to further investigate the relationship between the modification factor F and any coding-derived feature that allowed a better modeling of ME-compromising situations by generating a more precise estimation of the optimal λ_i^*.

In HEVC, a more precise modeling of F_{opt} by evaluating the best choice on a frame-basis $F_{opt}^{(k)}$ (instead of sequence-basis) would allow us to better understand the reasons why the reference $\lambda(QP)$ relationship fails and to find better features for the classification stage, which has been determined to be critical. Also, other more sophisticated classification tools could be evaluated.

Bibliography

[Altunbasak and Kamaci, 2004] Altunbasak, Y. and Kamaci, N. (2004). An Analysis of the DCT Coefficient Distribution with the H.264 Video Coder. In *Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP '04). IEEE International Conference on*, volume 3, pages iii–177–80 vol.3.

[Biatek et al., 2014] Biatek, T., Raulet, M., Travers, J.-F., and Deforges, O. (2014). Efficient Quantization Parameter Estimation in HEVC Based on ρ-domain. In *Signal Processing Conference (EUSIPCO), 2013 Proceedings of the 22nd European*, pages 296–300. IEEE.

[Bishop et al., 2006] Bishop, C. M. et al. (2006). *Pattern Recognition and Machine Learning*, volume 4-4. Springer New York.

[Bjontegaard, 2001] Bjontegaard, G. (2001). Calculation of Average PSNR Differences between RD-Curves, VCEG-M33. *ITU-Telecommunications Standardization Secto*, pages 290–294.

[Bossen, 2013] Bossen, F. (2013). Common Test Conditions and Software Reference Configurations. JCTVC-L1100.

[Bossen et al., 2013] Bossen, F., Flynn, D., and Sühring, K. (2013). High Efficiency Video Coding (HEVC) Test Model 12 (HM 12) Reference Software.

[Boyce, 2004] Boyce, J. (2004). Weighted Prediction in the H.264/MPEG AVC Video Coding Standard. In *Circuits and Systems, 2004. ISCAS '04. Proceedings of the 2004 International Symposium on*, volume 3, pages III – 789–92 Vol.3.

[Budagavi, 2005] Budagavi, M. (2005). Video Compression using Blur Compensation. In *Image Processing, 2005. ICIP 2005. IEEE International Conference on*, volume 2, pages II – 882–5.

[Channappayya et al., 2008] Channappayya, S., Bovik, A., and Heath, R. (2008). Rate Bounds on SSIM Index of Quantized Images. *Image Processing, IEEE Transactions on*, 17(9):1624–1639.

[Chen and Garbacea, 2006] Chen, L. and Garbacea, I. (2006). Adaptive Lambda Estimation in Lagrangian Rate-Distortion Optimization for Video Coding. In *Visual Communications and Image Processing 2006*, volume 6077-1, page 60772B. SPIE.

[Chiang and Zhang, 1997] Chiang, T. and Zhang, Y.-Q. (1997). A New Rate Control Scheme Using Quadratic Rate Distortion Model. *Circuits and Systems for Video Technology, IEEE Transactions on*, 7(1):246 –250.

[Dai et al., 2014] Dai, W., Au, O., Zhu, W., Wan, P., Hu, W., and Zhou, J. (2014). SSIM-based Rate-Distortion Optimization in H.264. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pages 7343–7347.

[de Frutos-López et al., 2015] de Frutos-López, M., González-de Suso, J. L., Sanz-Rodríguez, S., Peláez-Moreno, C., and Díaz-de María, F. (2015). Two-Level Sliding-Window VBR Control Algorithm for Video on Demand Streaming. *Signal Processing: Image Communication*, 36:1–13.

[Deng et al., 2013] Deng, L., Pu, F., Hu, S., and Kuo, C.-C. J. (2013). HEVC Encoder Optimization Based on a New RD Model and Pre-Encoding. In *Picture Coding Symposium (PCS 2013)*. IEEE.

[Ding and Liu, 1996] Ding, W. and Liu, B. (1996). Rate Control of MPEG Video Coding and Recording by Rate-Quantization Modeling. *Circuits and Systems for Video Technology, IEEE Transactions on*, 6(1):12 –20.

[Everett, 1963] Everett, H. (1963). Generalized Lagrange Multiplier Method for Solving Problems of Optimum Allocation of Resources. *Operations Research*, 11(3):399–417.

[Gish and Pierce, 1968] Gish, H. and Pierce, J. (1968). Asymptotically Efficient Quantizing. *Information Theory, IEEE Transactions on*, 14(5):676–683.

[González-de Suso et al., 2014] González-de Suso, J. L., Jiménez-Moreno, A., Martínez-Enríquez, E., and Díaz-de María, F. (2014). Improved Method to Select the Lagrange Multiplier for Rate-Distortion Based Motion Estimation in Video Coding. *Circuits and Systems for Video Technology, IEEE Transactions on*, 24(3):452–464.

[González-de Suso et al., 2016] González-de Suso, J. L., Martínez-Enríquez, E., and Díaz-de María, F. (2016). Adaptive Lagrange Multiplier Estimation Algorithm in HEVC. *Multimedia, IEEE Transactions on*. Submitted.

[Hang and Chen, 1997] Hang, H.-M. and Chen, J.-J. (1997). Source Model for Transform Video Coder and its Application. I. Fundamental Theory. *Circuits and Systems for Video Technology, IEEE Transactions on*, 7(2):287 –298.

[He and Mitra, 2002] He, Z. and Mitra, S. (2002). Optimum Bit Allocation and Accurate Rate Control for Video Coding via ρ-domain Source Modeling. *Circuits and Systems for Video Technology, IEEE Transactions on*, 12(10):840–849.

[ISO/IEC, 1993] ISO/IEC (1993). Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to 1.5 Mbit/s-Part 2: Video, ISO/IEC 11172-2 (MPEG-1).

[ISO/IEC, 1999] ISO/IEC (1999). Coding of Audio-Visual Objects - Part 2: Visual, ISO/IEC 14496-2 (MPEG-4 Visual version 1). (and subsequent editions).

[ITU-T, 1990] ITU-T (1990). H.261-Video Codec for Audiovisual Services at px64 kbit/s.

[ITU-T, 1995] ITU-T (1995). Video Coding for Low Bit Rate Communications, ITU-T Rec. H.263. (and subsequent editions).

[ITU-T and ISO/IEC, 1994] ITU-T and ISO/IEC (1994). Generic Coding of Moving Pictures and Associated Audio Information - Part 2: Video, ITU-T Rec. H.262 and ISO/IEC 13818-2 (MPEG-2 Video).

[Jiménez-Moreno et al., 2016] Jiménez-Moreno, A., Martínez-Enríquez, E., and de María, F. D. (2016). Complexity Control Based on a Fast Coding Unit Decision Method in the HEVC Video Coding Standard. *IEEE Transactions on Multimedia*, PP(99):1–1.

[JVT, 2003] JVT (2003). Advanced Video Coding (AVC), ITU-T Rec. H.264 and ISO/IEC 14496-10 (MPEG-4 Part 10).

[JVT, 2010] JVT (2010). H.264/AVC Reference Software v15.1.

[JVT, 2013] JVT (2013). High Efficiency Video Coding (HEVC), ITU-T H.265 and ISO/IEC 23008-2. 1 ed.

[Kamaci and Altunbasak, 2005] Kamaci, N. and Altunbasak, Y. (2005). Frame Bit Allocation for H.264 Using Cauchy-Distribution Based Source Modelling. In *Acoustics, Speech, and Signal Processing, 2005. Proceedings. (ICASSP '05). IEEE International Conference on*, volume 2, pages 57–60.

[Kamikura et al., 1998] Kamikura, K., Watanabe, H., Jozawa, H., Kotera, H., and Ichinose, S. (1998). Global Brightness-Variation Compensation for Video Coding. *Circuits and Systems for Video Technology, IEEE Transactions on*, 8(8):988 –1000.

[Kim et al., 2012] Kim, I.-K., McCann, K., Sugimoto, K., Bross, B., and Han, W.-J. (2012). Hm9: High Efficiency Video Coding (HEVC) Test Model 9 Encoder Description. In *9th JCT-VC Meeting, Switzerland*, pages 10–11.

[Lam and Goodman, 2000] Lam, E. and Goodman, J. (2000). A Mathematical Analysis of the DCT Coefficient Distributions for Images. *Image Processing, IEEE Transactions on*, 9(10):1661–1666.

[Le Pennec and Mallat, 2005] Le Pennec, E. and Mallat, S. (2005). Sparse Geometric Image Representations with Bandelets. *Image Processing, IEEE Transactions on*, 14(4):423 –438.

[Lee and Kim, 2011] Lee, B. and Kim, M. (2011). Modeling Rates and Distortions Based on a Mixture of Laplacian Distributions for Inter-Predicted Residues in Quadtree Coding of HEVC. *Signal Processing Letters, IEEE*, 18(10):571–574.

[Li et al., 2014] Li, B., Li, H., Li, L., and Zhang, J. (2014). λ Domain Rate Control Algorithm for High Efficiency Video Coding. *Image Processing, IEEE Transactions on*, 23(9):3841–3854.

[Li et al., 2015] Li, S., Zhu, C., Gao, Y., Zhou, Y., Dufaux, F., and Sun, M. (2015). Lagrangian Multiplier Adaptation for Rate-Distortion Optimization with Inter-frame Dependency. *Circuits and Systems for Video Technology, IEEE Transactions on*, PP(99):1–1.

[Li et al., 2009] Li, X., Oertel, N., Hutter, A., and Kaup, A. (2009). Laplace Distribution Based Lagrangian Rate Distortion Optimization for Hybrid Video Coding. *Circuits and Systems for Video Technology, IEEE Transactions on*, 19(2):193 –205.

[Lim et al., 2005] Lim, K.-P., Sullivan, G., and Wiegand, T. (2005). Text Description of Joint Model Reference Encoding Methods and Decoding Concealment Methods. JVT-0079.doc.

[Liu et al., 2012] Liu, Z., Wang, D., Zhou, J., and Ikenaga, T. (2012). Lagrangian Multiplier Optimization Using Correlations in Residues. In *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pages 1185–1188.

[Ma et al., 2012] Ma, S., Si, J., and Wang, S. (2012). A Study on the Rate Distortion Modeling for High Efficiency Video Coding. In *Image Processing (ICIP), 2012 19th IEEE International Conference on*, pages 181–184. IEEE.

[Martínez-Enríquez et al., 2010] Martínez-Enríquez, E., Jiménez-Moreno, A., and Díaz-de María, F. (2010). An Adaptive Algorithm for Fast Inter Mode Decision in the H. 264/AVC Video Coding Standard. *Consumer Electronics, IEEE Transactions on*, 56(2):826–834.

[McCann et al., 2014] McCann, K., Rosewarne, C., Bross, B., Naccari, M., Sharman, K., and Sullivan, G. (2014). High Efficiency Video Coding (HEVC) Test Model 16 (HM 16) Improved Encoder Description.

[Molinero et al., 2011] Molinero, J., Jiménez, A., Martínez-Enríquez, E., and Díaz-de María, F. (2011). On the Optimal Lagrangian Parameter for Motion Estimation: A Low-Cost and Effective Method for Improving Video Coding Performance. In *Telecommunications (CONATEL), 2011 2nd National Conference on*, pages 1–6.

[Ortega and Ramchandran, 1998] Ortega, A. and Ramchandran, K. (1998). Rate-Distortion Methods for Image and Video Compression. *Signal Processing Magazine, IEEE*, 15(6):23 –50.

[Ramchandran and Vetterli, 1993] Ramchandran, K. and Vetterli, M. (1993). Best Wavelet Packet Bases in a Rate-Distortion Sense. *Image Processing, IEEE Transactions on*, 2(2):160 –175.

[Rehman and Wang, 2012] Rehman, A. and Wang, Z. (2012). SSIM-inspired Perceptual Video Coding for HEVC. In *Multimedia and Expo (ICME), 2012 IEEE International Conference on*, pages 497–502. IEEE.

[Richardson, 2003] Richardson, I. (2003). *H.264 and MPEG-4 Video Compression: Video Coding for Next Generation Multimedia*. John Wiley & Sons.

[Sangi et al., 2004] Sangi, P., Heikkila, J., and Silven, O. (2004). Selection of the Lagrange multiplier for block-based motion estimation criteria. In *Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP '04). IEEE International Conference on*, volume 3, pages iii – 325–8 vol.3.

[Schwarz et al., 2006] Schwarz, H., Marpe, D., and Wiegand, T. (2006). Analysis of hierarchical b pictures and mctf. In *Multimedia and Expo, 2006 IEEE International Conference on*, pages 1929–1932.

[Shen et al., 2013] Shen, L., Liu, Z., Zhang, X., Zhao, W., and Zhang, Z. (2013). An Effective CU Size Decision Method for HEVC Encoders. *IEEE Transactions on Multimedia*, 15(2):465–470.

[Shoham and Gersho, 1988] Shoham, Y. and Gersho, A. (1988). Efficient Bit Allocation for an Arbitrary Set of Quantizers [Speech Coding]. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 36(9):1445 –1453.

[Si et al., 2013] Si, J., Ma, S., Wang, S., and Gao, W. (2013). Laplace Distribution Based CTU Level Rate Control for HEVC. In *Visual Communications and Image Processing (VCIP), 2013*, pages 1–6. IEEE.

[Sullivan, 2001] Sullivan, G. (2001). Recommended simulation common conditions for h. 26l coding efficiency experiments on low resolution progressive scan source material. *ITU-T VCEG-N81, September 24-27, 2001*, pages 24–27.

[Sullivan and Wiegand, 1998] Sullivan, G. and Wiegand, T. (1998). Rate-Distortion Optimization for Video Compression. *Signal Processing Magazine, IEEE*, 15(6):74 –90.

[Sullivan et al., 2012] Sullivan, G. J., Ohm, J., Han, W.-J., and Wiegand, T. (2012). Overview of the High Efficiency Video Coding (HEVC) Standard. *Circuits and Systems for Video Technology, IEEE Transactions on*, 22(12):1649–1668.

[Wang et al., 2013] Wang, S., Ma, S., Wang, S., Zhao, D., and Gao, W. (2013). Quadratic ρ-domain Based Rate Control Algorithm for HEVC. In *Acoustics, Speech*

and Signal Processing (ICASSP), 2013 IEEE International Conference on, pages 1695–1699.

[Wang et al., 2011] Wang, S., Rehman, A., Wang, Z., Ma, S., and Gao, W. (2011). Rate-SSIM Optimization for Video Coding. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pages 833–836.

[Wang et al., 2004] Wang, Z., Bovik, A., Sheikh, H., and Simoncelli, E. (2004). Image quality assessment: from error visibility to structural similarity. *Image Processing, IEEE Transactions on*, 13(4):600–612.

[Wiegand and Girod, 2001] Wiegand, T. and Girod, B. (2001). Lagrange Multiplier Selection in Hybrid Video Coder Control. In *Image Processing, 2001. Proceedings. 2001 International Conference on*, volume 3, pages 542 –545 vol.3.

[Wiegand et al., 2003a] Wiegand, T., Schwarz, H., Joch, A., Kossentini, F., and Sullivan, G. (2003a). Rate-Constrained Coder Control and Comparison of Video Coding Standards. *Circuits and Systems for Video Technology, IEEE Transactions on*, 13(7):688–703.

[Wiegand et al., 2003b] Wiegand, T., Sullivan, G., Bjontegaard, G., and Luthra, A. (2003b). Overview of the H.264/AVC Video Coding Standard. *Circuits and Systems for Video Technology, IEEE Transactions on*, 13(7):560–576.

[Wu and Gersho, 1991] Wu, S.-W. and Gersho, A. (1991). Rate-Constrained Optimal Block-Adaptive Coding for Digital Tape Recording of HDTV. *Circuits and Systems for Video Technology, IEEE Transactions on*, 1(1):100 –112.

[Xiong et al., 2014] Xiong, J., Li, H., Wu, Q., and Meng, F. (2014). A Fast HEVC Inter CU Selection Method Based on Pyramid Motion Divergence. *IEEE Transactions on Multimedia*, 16(2):559–564.

[Yeo et al., 2013] Yeo, C., Tan, H. L., and Tan, Y. H. (2013). On Rate Distortion Optimization Using SSIM. *Circuits and Systems for Video Technology, IEEE Transactions on*, 23(7):1170–1181.

[Yovanof and Liu, 1996] Yovanof, G. and Liu, S. (1996). Statistical Analysis of the DCT Coefficients and their Quantization Error. In *Signals, Systems and Computers, 1996. Conference Record of the Thirtieth Asilomar Conference on*, volume 1, pages 601–605 vol.1.

[Zeng et al., 2013] Zeng, H., Ngan, K. N., and Wang, M. (2013). Perceptual Adaptive Lagrangian Multiplier for High Efficiency Video Coding. In *Picture Coding Symposium (PCS), 2013*, pages 69–72.

[Zhang et al., 2010] Zhang, J., Yi, X., Ling, N., and Shang, W. (2010). Context Adaptive Lagrange Multiplier (CALM) for Rate-Distortion Optimal Motion Estimation in Video Coding. *Circuits and Systems for Video Technology, IEEE Transactions on*, 20(6):820 –828.

[Zhang et al., 2014] Zhang, X., Tian, Y., Huang, T., Dong, S., and Gao, W. (2014). Optimizing the Hierarchical Prediction and Coding in HEVC for Surveillance and Conference Videos with Background Modeling. *IEEE Transactions on Image Processing*, 23(10):4511–4526.

[Zhao et al., 2013] Zhao, L., Zhang, X., Tian, Y., Wang, R., and Huang, T. (2013). A Background Proportion Adaptive Lagrange Multiplier Selection Method for

Surveillance Video on HEVC. In *Multimedia and Expo (ICME), 2013 IEEE International Conference on*, pages 1–6. IEEE.